Structure Determination of Membrane Proteins by Electron Crystallography

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Andreas Daniel Schenk

aus Signau BE

Basel, 2006

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Andreas Engel und Prof. Dr. Ueli Aebi

Basel, den 14.02.2006

Prof. Dr. Hans-Jakob Wirz
Dekan

Acknowledgment

I wish to thank Ansgar Philippsen for the fruitful discussions and his support in all questions of image processing. I'm thankful to Henning Stahlberg and Paul Werten who guided me trough the aquaporin-2 project. I would also like to thank Simon Scheuring and Patrick Frederix for the nice AFM pictures of AQP2 and Shirley Müller and Vesna Olivieri who took the STEM images for the mass measurement and which also helped me analyse this data. I would also like to thank Bert de Groot who fitted the helical fragments into the AQP2 density and compared the helix axis tilts to AQP1.

Thanks also go to Giani Signorell, Hervé Remigy and Mohamed Chami for the good collaboration on the KdgM project.

I especially would like to thank Andreas Engel who was my supervisor during the PhD thesis and gave me the opportunity to work in his Laboratory on this very interesting project.

Contents

1 **Introduction** **7**

2 **Structural Investigations of Membrane Proteins** **9**

- 2.1 Overview . . . 9
  - 2.1.1 Aquaporins . . . 9
  - 2.1.2 Porins . . . 10
- 2.2 The 4.5 Å Structure of Human AQP2 . . . 11
  - 2.2.1 Summary . . . 11
  - 2.2.2 Addendum . . . 14
  - 2.2.3 Outlook . . . 16
- 2.3 The 5 Å Structure of Heterologously Expressed Plant Aquaporin SoPIP2;1 29
  - 2.3.1 Summary . . . 29
  - 2.3.2 Addendum . . . 30
- 2.4 Oligogalacturonate Porin KdgM . . . 37
  - 2.4.1 Introduction . . . 37
  - 2.4.2 Image Processing . . . 39
  - 2.4.3 Subunit Composition . . . 39
  - 2.4.4 Outlook . . . 43

3 **New Developments in Image Processing** **46**

- 3.1 Introduction . . . 46
  - 3.1.1 Processing of Cryo-EM Images . . . 46
  - 3.1.2 Processing of Electron Diffraction Patterns . . . 49

3.1.3 3D Reconstruction 49
3.1.4 Problems encountered 51
3.2 New Developments in Processing Cryo Images 53
3.2.1 Spotlist 53
3.2.2 Image Preparation 53
3.2.3 Tilt Geometry 54
3.2.4 Unbending 57
3.2.5 Masking 58
3.2.6 Projection Image Generation 60
3.2.7 Phase Origin 60
3.2.8 Symmetrization in 2D 62
3.3 New Developments in Processing of Electron Diffraction Data 62
3.3.1 Data Extraction 63
3.3.2 Origin Determination 63
3.4 General Developments Concerning Image and Diffraction Processing . . 65
3.4.1 Increased Greyscale Depth 65
3.4.2 Linux Port 67
3.4.3 Standardized Directory Structure 68
3.4.4 Directory Walker 70
3.4.5 Load Balancing 70
3.4.6 Storage Handling 70
3.4.7 Image Display with XIMDISP 71
3.4.8 Lattice Determination 72
3.4.9 Reindexing 77
3.4.10 Merging 78
3.4.11 Discretization 79
3.4.12 Lattice Line Prediction 79
3.4.13 Projects Statistics and Consistency Tests 79
3.4.14 Interoperability 83

3.4.15 Meta Info 84
3.4.16 GUI 89
3.4.17 Checkpointing 89
3.5 IPLT–Image Processing Library and Toolkit 90
3.5.1 Summary 90
3.5.2 Addendum 91

4 Improvements in Sample Preparation and Image Acquisition 101
4.1 Specimen Preparation 102
4.1.1 Trehalose Embedding 102
4.1.2 Glucose Embedding 103
4.2 Image and Electron Diffraction Acquisition 103
4.2.1 Calibration 103
4.2.2 Spot Scan Imaging 106
4.2.3 Image Digitalization 108
4.3 Electron Diffraction 108
4.3.1 Acquisition 108

5 Conclusion 110

List of Figures

2.1 Vassopressin induced regulation of AQP2 12

2.2 Aquaporin 2 secondary structure 13

2.3 Sampling density in Fourier space 15

2.4 Diffraction lattice line plots 15

2.5 Tubular crystal of KdgM 38

2.6 Fourier transform of figure 2.5 40

2.7 Averaged unit cell of KdgM 42

2.8 Secondary structure prediction 44

2.9 Rotational power spectrum 45

3.1 Processing scheme for electron crystallographic images 47

3.2 Processing scheme for electron diffraction patterns 50

3.3 Fitted defocus values 56

3.4 Displacement vectors 57

3.5 Reference images created by MAKETRAN and IPLT 59

3.6 Mask generation by morphological algorithms 61

3.7 Gauss fit of a diffraction peak 64

3.8 Peak integration 65

3.9 Explicit gauss fit of the background 66

3.10 Standard directory structure 69

3.11 Ximdisp showing an overlay for interactive CTF fitting 73

3.12 Index hinting 74

3.13 Storing of an indexed lattice to a meta file 75
3.14 Spotlist generation 76
3.15 The iplt lattice viewer 78
3.16 Tilt angle plot 81
3.17 Axis angle plot 82
3.18 Building blocks of the image processing framework 85
3.19 Hierarchical XML structure 86

4.1 Humidity chamber 104
4.2 Hydraulic loop holder 105
4.3 Scheme of a trehalose embedded crystal 106
4.4 Magnification calibration for the CM200FEG 107
4.5 Beam stop modification 109

List of Tables

2.1 Phase residual table for KdgM 41

2.2 Crystallographic data of KdgM 41

Chapter 1

Introduction

A fundamental principle of life is the separation of environments into different compartments. Prokaryotes shield their interior from the environment by a plasma membrane and in some cases also by a cell wall. Eukaryotes refine this compartmentalization by building different organelles for different parts of the cell metabolism. Nevertheless, these different compartments are dependent on each other and are interconnected by membrane proteins that transport specific nutrients, hormones, ions, water and waste products across the membrane and facilitate signal transmission between different compartments. Understanding the structure and function of membrane proteins can therefore allow an enormous insight into the regulation of different metabolic pathways.

The electron microscope (EM) proved itself a great tool for studying membrane proteins, offering the unique opportunity to image membrane proteins within a lipid bilayer as close to the natural conditions as possible. Processing of images acquired by an electron microscope poses a challenging task for both scientist and processing hardware. Newly developed and optimized algorithms are needed to improve the image processing to a level that allows atomic resolution to be achieved regularly.

Membrane proteins pose a difficult challenge for a structural biologist. To crystallize membrane proteins into well ordered two dimensional (2D) or three dimensional (3D) crystals is one of the most important prerequisites for structural analysis at the atomic level, yet membrane proteins are notoriously difficult to crystallize.

One exception may be bacteriorhodopsin, which forms near-perfect crystals already in its native membrane. This may explain the fact that the first 2D electron crystallographic structure determined at 7 Å resolution by Henderson and Unwin[20][43] in 1975 was the structure of bacteriorhodopsin. In 1990 the structure of Br was determined to atomic resolution by Henderson et al.[19], being the first atomic structure of a membrane protein. The structure determination of Br was also the starting point for the MRC program suite, which is widely used at the moment in the, albeit small, 2D electron crystallography community. Using the MRC software Kühlbrandt et al.[26] solved the structure of the light-harvesting chlorophyll a/b-protein complex in 1994. For recording the images they used the spot scan technique developed by Downing in 1991[9].

The first aquaporin water channel determined was aquaporin 1, resolved by Walz et al. in 1997[45] at 6 Å resolution, and subsequently solved to atomic resolution by Murata et al. in 2000[29]. Recently, several more aquaporin structures were determined by 2D electron crystallographic methods, aquaporin-0 (AQP0) by Gonen et al. in 2004[14] at 3 Å and in 2005[13] at 1.9 Å and aquaporin-4 (AQP4) by Hiroaki et al. in 2006[22]. Interestingly, AQP4 shows exactly the same monomer arrangement as SoPIP2;1. The recent publications show that the trend goes from recording solely images to the recording of diffraction data in combination with images or even to recording diffraction data exclusively, and then using methods developed for x-ray crystallography to obtain the phase information.

Given the fact that the software available for processing of 2D electron diffraction patterns is less evolved than the one for processing images, and given this new development of increased usage of diffraction patterns, it only makes sense to focus on implementing new and improved programs for 2D electron diffraction processing.

In this work I would like to present the advances I achieved in the structural determination of aquaporin 2, as well as my contribution to other projects, in particular the structural investigations of SoPIP2;1 and KdgM. I will also explain the modified sample preparation methods which made data recording at high tilt angles more reliable and achieved an improvement in resolution of the measured data.

A second, equally important and detailed part of my thesis is the work invested in improving and extending the image processing to a point where a user, not adept in programming in several languages, can use it and produce good results. For this I improved the functionality and performance at several points, including a strong emphasis on user friendliness and ease of maintenance.

Chapter 2

Structural Investigations of Membrane Proteins

2.1 Overview

2.1.1 Aquaporins

Aquaporins belong to an ubiquitous family of membrane proteins. They facilitate the efficient permeation of water across the plasma membrane of cells in all living organism. Since the diffusion of water molecules through lipid bilayers has an activation energy of larger than $10\frac{kcal}{mol}$ [6], the existence of specific water pores was postulated more than four decades ago [41]. The first of these pores was found by Preston in 1992[37].

Mammalian aquaporins are members of a large family of pore-forming membrane proteins, the MIP family. The MIP family is divided into two subfamilies: the glyceroporins and the aquaporins. The glyceroporins (Glps) are channels for glycerol and other small nonionic solutes. They normally have a low permeability for water. On the other side, the aquaporins (AQPs) have a high permeability for water but they exclude ions and solutes like glycerol. In mammals some members of the AQP family are expressed in most tissues. AQP2, for instance, is predominantly found in the principal cells of the renal collecting duct, where it is responsible for the reabsorbtion of water from the urine.

Efficient permeation of water across the plasma membrane is also important for cytosolic osmoregulation in plants. Proteins of the aquaporin family are key components in cellular water homeostasis and they account for a significant fraction of the total amount of integral membrane proteins of plant plasma membranes. Their importance is also demonstrated by the fact that in Arabidopsis the expression products of 35 genes are aquaporin-like proteins and around one third of these are located at the plasma membrane.

At the cellular level the maintenance of the water balance is an interplay between plasma membrane and tonoplast aquaporins (PIPs and TIPs: Plasma membrane Intrinsic Proteins and Tonoplast Intrinsic Proteins, respectively). The PIPs are subdivided into two groups, the PIP1 and PIP2 isoforms. The latter have a longer C-terminal region and when expressed in Xenopus oocytes, show a higher water transport activity than the PIP1 isoforms[7]. SoPIP2;1, in previous nomenclature called PM28A, is a PIP2 isoform in *Spinacia oleracea* (spinach) leaf plasma membranes.

2.1.2 Porins

Gram negative bacteria are protected by an outer membrane against harsh environments. Nevertheless, translocation of solutes and proteins through the outer membrane has to be allowed, as they are crucial to the bacterial cell. Exchange of small molecules and ions is facilitated by members of the porin family, which are β-barrel proteins inserted into the outer membrane, forming small, water filled channels, allowing diffusion of small molecules and ions[25].They are divided into two classes: i) the non-specific porins of the general bacterial porin family, such as OmpF and OmpC of Escherichia coli, permitting diffusion of molecules below about 600 Da[31]; ii) the substrate-specific porins such as LamB[4] facilitating the diffusion of specific substrates like complex sugars.

KdgM is a *Erwinia chrysanthemi* oligogalacturonate-specific monomeric porin which does not have detectable homology with any porin of known structure. Based upon sequence similarity and the amphipathy profile, a model featuring a β-barrel composed of 14 antiparallel β-strands was constructed[33]. KdgM is involved in the transport of oligogalacturonates degraded by secreted pectinases through the outer membrane to the periplasma. From there the oligogalacturonates are the transported to the cytosol by members of the carbohydrate uptake transporter type 1 family, where they are used by *Erwinia chrysanthemi* as a carbon source for growth[3].

2.2 The 4.5 Å Structure of Human AQP2

Contribution

The following work was published in 2005 in the Journal of Molecular Biology Volume 350 on the pages 278-289. The work presented here is the main project of my PhD thesis and I am the main author of this publication.

My contribution to the project consisted in the purification and crystallization of AQP2 with the help of Paul Werten, the TEM data acquisition, the TEM data processing and the modification of the processing software initially assisted by Henning Stahlberg, the STEM data evaluation and mass per area determination with the help of Shirley Müller, the coding of the layer separation algorithm, the fitting of the AQP1 structure dataset into the AQP2 mass density helped by Bert de Groot. The AFM images were recorded by Simon Scheuring and the STEM microscopy for the mass measurements was done by Vesna Olivieri. The helix-fitting was carried out by Bert de Groot.

2.2.1 Summary

Aquaporin 2 is a mammalian water channel predominantly found in the apical membrane of the principal cells of the renal collection duct, although it is also expressed in other tissues.

Distribution of AQP2 between intracellular storage vesicles and the apical membrane is controlled by the anti- diuretic hormone vasopressin (AVP)[30][24][8]. (Fig. 2.1). Recent findings show that soluble N-ethylmaleimide sensitive fusion factor attachment protein receptors (SNARE) and actin cytoskeleton organization, regulated by a small GTPase of the Rho family, are also essential for AQP2 trafficking[44][32].

AQP2 is, as all aquaporins, a homo-tetrameric membrane protein, each monomer comprising six transmembrane helices (Fig. 2.2), forming a right-handed bundle that houses an independent channel.

AQP2 crystallized into double layered 2D crystals, formed by two p4 symmetric layers, which exposes both extracellular sides to the surrounding and buries the cytosolic sides, containing N- and C-terminus, within the crystal. In initial crystallization setups, different register shifts between the two layers could be observed, but optimising the crystallization conditions yielded crystals with a constant register shift of half a unit cell, in either x- or y-direction, between the two layers. Although the double layer apparently improved the crystallinity, it complicated image processing tremendously.

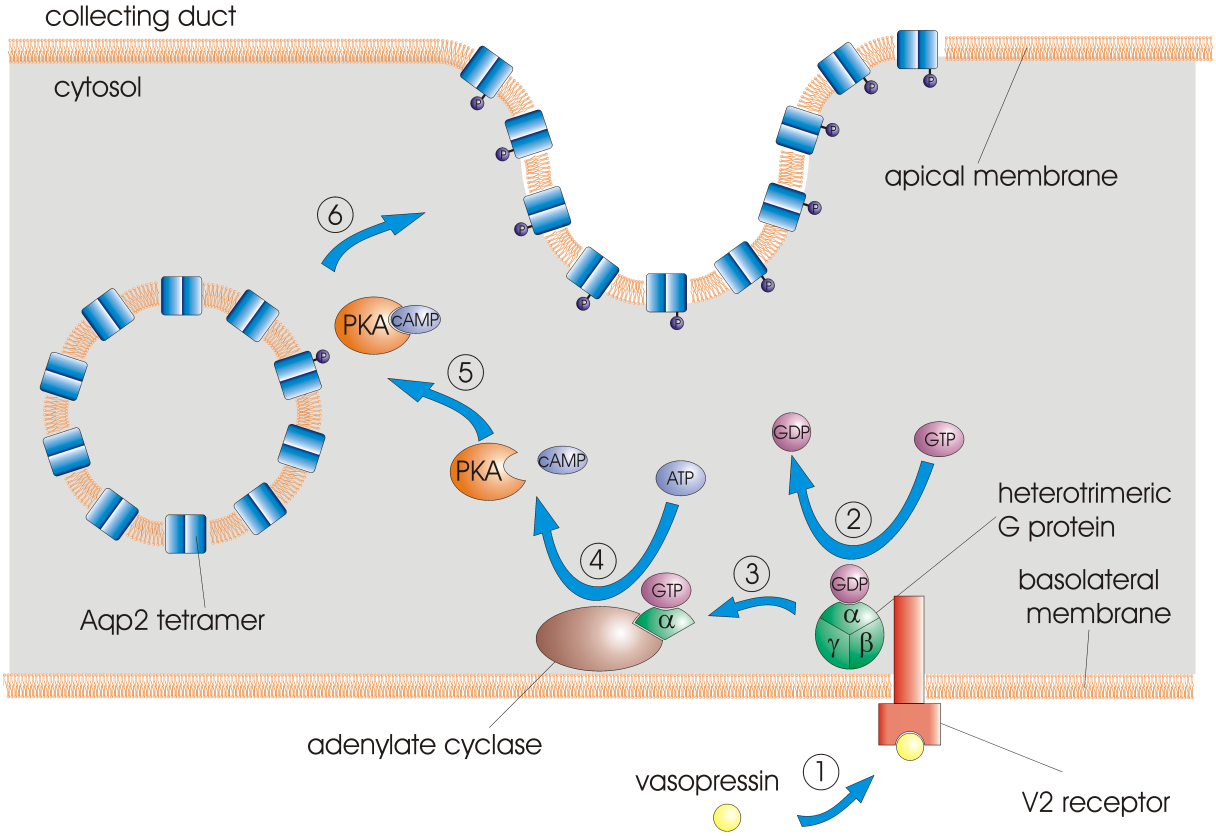


Figure 2.1: Vassopressin induced regulation of AQP2 (1)Vasopressin binds to the V2-receptor. (2)The activated V2-receptor binds to the G protein and activates it. The activation of the G protein leads to the release of GDP and binding of GTP. (3)The α subunit dissociates from the $\beta - \gamma$ complex. (4)Binding of α to the adenylate cyclase activates the synthesis of cAMP from ATP. (5)cAMP triggers the phosphorylation of AQP2 by phosphokinase A.(6)The phosphorylation results in the redistribution of the AQP2 tetramers from intracellular storage vesicles to the apical membrane.

On the projection map, the individual tetramers could hardly be identified due to overlap. The overall symmetry of the double layer turned out to be $p22_12$, which had the unanticipated effect of having two unit cell vectors of the same length which were indistinguishable at indexing time. In addition, the register shift of half a unit cell proved to be a problem during unbending, because the cross correlation signal between the two layers was quite high, even though they were rotated against each other by 180° around the y-axis, leading to unwanted half unit cell shifts during unbending.

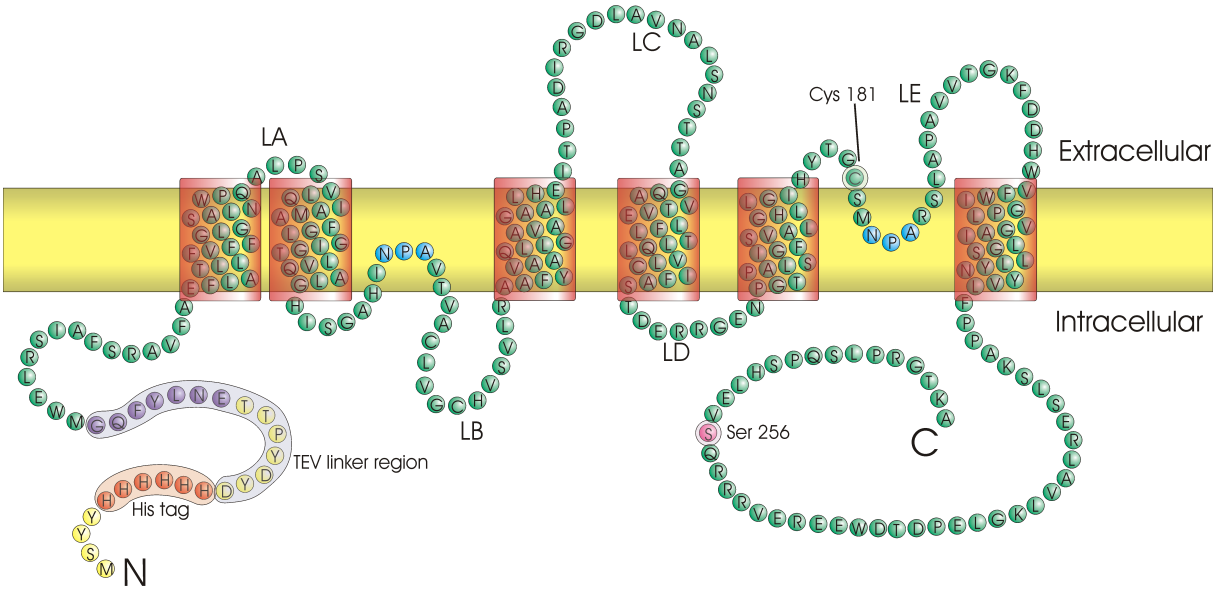


Figure 2.2: Aquaporin 2 secondary structure. A poly histidine tag has been added. It is connected to the native N terminus with a TEV protease cleavable linker region to facilitate protein purification. Serin 256 is the site of phosphorylation. Loop B and E containing the NPA motif project back into the membrane and form a seventh pseudo helix.

2.2.2 Addendum

Imaging of diffraction patterns: To record electron diffraction patterns the AQP2 crystals were embedded as described in 4.3. The patterns were recorded on a Gatan 2k×2k CCD camera as described in detail in 4.3.1. In total 470 diffraction patterns were recorded, 62 untilted, 61 at 10°, 116 at 15°, 30 at 20°, 61 at 30°, 117 at 45° and 24 at 60°.

Diffraction Data Processing: From the 470 patterns recorded the best 170 (29 untilted, 38 at 10°, 58 at 15°, 14 at 30°, and 31 at 45°) were processed. The sampling in Fourier space of the complete dataset can be seen in figure 2.3.

The diffraction patterns were processed using the algorithms implemented in IPLT as described in 3.3. The diffraction data was merged concurrently using both implemented approaches. As reference a merged AQP2 image dataset was taken. The merged diffraction dataset contains 77847 reflections. Afterwards the dataset generated using the common-line scaling was compared to the dataset generated by scaling against a global reference structure. Remarkable was the difference in scaling speed observed. Using a global reference structure the dataset was scaled within several minutes on a standard workstation, as opposed to the scaling using the common-line approach, which took more than one day. As figure 2.4 shows scaling using the common-line approach proved to be more reliable.

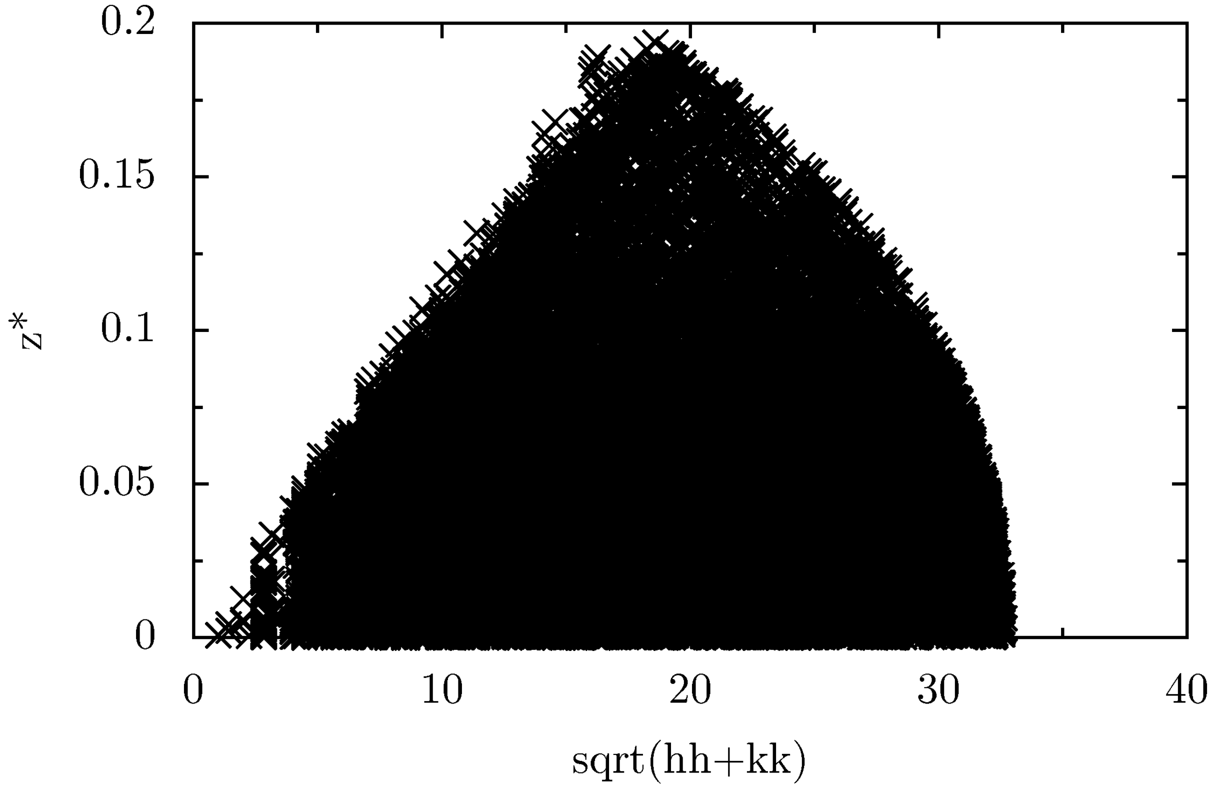


Figure 2.3: Sampling density in Fourier space. The graph shows the sampling of the Fourier space for the merged AQP2 diffraction dataset up to 45°.

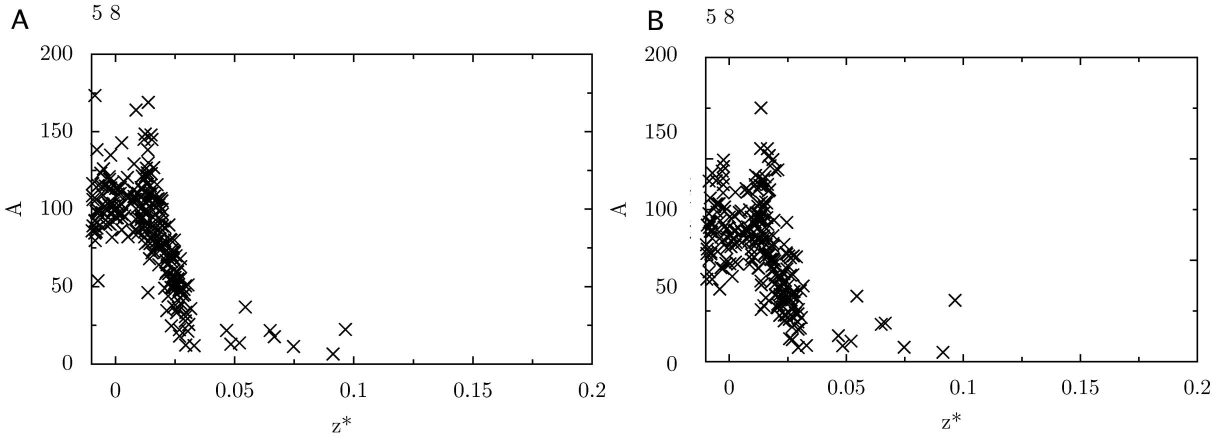


Figure 2.4: Diffraction lattice line plots. (A) Plot of the lattice line (5,8) determined using common-line scaling. (B) Plot of the lattice line (5,8) scaled using a reference map.

2.2.3 Outlook

Now that we have a merged dataset which samples a good portion of the Fourier space, the next step will be to discretize the dataset. Once the dataset has been discretized, one can think of several further strategies. One possibility will be to exploit the fact that the exact crystal arrangement is known from processing of the images and that atomic structures of similar proteins are available, with usage of the tools established in the X-ray community to solve the structure by molecular replacement. Another way will be to combine the diffraction amplitudes with the phases determined from the images and then use methods for phase extension to increase resolution.

doi:10.1016/j.jmb.2005.04.030
J. Mol. Biol. (2005) **350**, 278–289

JMB
Available online at www.sciencedirect.com
SCIENCE @ DIRECT®
ELSEVIER

The 4.5 Å Structure of Human AQP2

Andreas D. Schenk[1], Paul J. L. Werten[1], Simon Scheuring[2]
Bert L. de Groot[3], Shirley A. Müller[1], Henning Stahlberg[4]
Ansgar Philippsen[1] and Andreas Engel[1]*

[1]*M. E. Müller Institute for Microscopy, Biozentrum University of Basel Klingelbergstrasse 70, 4056 Basel, Switzerland*

[2]*Institut Curie, UMR-CNRS 168 and LRC-CEA 34V, 11 rue Pierre et Marie Curie, 75231 Paris Cedex 05, France*

[3]*Computational Biomolecular Dynamics Group Max-Planck-Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen Germany*

[4]*Molecular and Cellular Biology, University of California, 1 Shields Avenue Davis, CA 95616, USA*

**Corresponding author*

Located in the principal cells of the collecting duct, aquaporin-2 (AQP2) is responsible for the regulated water reabsorbtion in the kidney and is indispensable for the maintenance of body water balance. Disregulation or malfunctioning of AQP2 can lead to severe diseases such as nephrogenic diabetes insipidus, congestive heart failure, liver cirrhosis and pre-eclampsia. Here we present the crystallization of recombinantly expressed human AQP2 into two-dimensional protein-lipid arrays and their structural characterization by atomic force microscopy and electron crystallography. These crystals are double-layered sheets that have a diameter of up to 30 μm, diffract to 3 Å^{-1} and are stacked by contacts between their cytosolic surfaces. The structure determined to 4.5 Å resolution in the plane of the membrane reveals the typical aquaporin fold but also a particular structure between the stacked layers that is likely to be related to the cytosolic N and C termini.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: aquaporin; atomic force microscopy; electron crystallography; nephrogenic diabetes insipidus; 2D crystallization

Introduction

As water is the major constituent of all forms of life, maintaining water homeostasis is crucial for all living organisms. While water passes through pure lipid bilayers with high activation energy ($E_a > 10$ kcal/mol), the rapid flow of water through specialized membranes, e.g. in the kidneys, occurs with an $E_a < 5$ kcal/mol. Members of the aquaglyceroporin family facilitate this efficient transport of water and small solutes across biological membranes.[1–5] Phylogenetic analyses have revealed two distinct subfamilies, the aquaporin (AQP) and the glyceroporin (GLP) subfamilies. Members of the former are highly specific for water, but members of the latter also permeate small solutes such as glycerol.[6] Aquaglyceroporins are homo-tetrameric membrane proteins, with each monomer comprising six transmembrane helices forming a right-handed bundle that houses an independent channel. Site-directed mutagenesis on the functional loops containing the NPA motifs has lead to the "hourglass" model in which the intracellular and the extracellular NPA loops project back into the membrane bilayer where their intersection forms a narrow aperture.[7] Two unusual half-helices in these loops emanate outwards from the two highly conserved stacked proline residues in the middle of the membrane. These half-helices form a pseudo seventh helix, and together with helices 1, 2, 4 and 5 build the framework for the actual channel.[8] Altogether six atomic mammalian aquaporin structures (human AQP1 (1FQY;[8] 1H6I;[9] 1IH5[10]), bovine AQP1 (1J4N[11]) and AQP0 (1SOR;[12] 1YMG[13])), and four structures of bacterial aquaglyceroporins (GlpF (1FX8;[14] 1LDA, 1LDI[15]) and AqpZ (1RC2[16])) have been deposited in the Protein Data Bank.

In the human kidney, several members of the

Abbreviations used: AQP2, aquaporin-2; HT, His-tagged; LPR, lipid-to-protein ratio; AFM, atomic force microscopy; MPA, mass-per-area; STEM, scanning transmission electron microscopy.

E-mail address of the corresponding author:
andreas.engel@unibas.ch

0022-2836/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.

aquaporin family are responsible for water reabsorbtion. Of the 180 liters of pro-urine filtered daily, 90% is reabsorbed *via* AQP1, which is expressed in the apical and basolateral membranes of epithelial cells in the proximal tubules and descending limbs of Henle. The remaining water reabsorbtion takes place in the collecting duct. It is mediated by AQP2 in the apical membrane and by AQP3 and AQP4 in the basolateral membrane of the principal cells of the collecting duct, and is controlled by the anti-diuretic hormone vasopressin (AVP).[17–19] Binding of AVP to its receptor (V2R) on the basolateral side of the collecting duct cells leads to activation of cAMP-dependent protein kinase A, which phosphorylates AQP2 at C-terminal residue Ser256. This promotes the redistribution of AQP2 from intracellular storage vesicles to the apical plasma membrane, where it can exert its function as a water channel. Removal of AVP reverses this process. Disregulation or malfunctioning of AQP2 can lead to a variety of severe diseases, such as nephrogenic diabetes insipidus[20–23], congestive heart failure,[23–26] liver cirrhosis[26–29] and pre-eclampsia,[23,26,30,31] showing the physiological importance of this protein.

The double-layered two-dimensional (2D) AQP2 crystals presented here diffract to 3 Å^{-1}. They were characterized by atomic force microscopy (AFM), transmission electron microscopy (TEM) and electron crystallography, and the structure of AQP2 was determined to a resolution of 4.5 Å in the plane of the membrane and 7 Å perpendicular to it. This 3D potential map was calculated using information from electron images only, and is the first medium resolution structure of a recombinantly expressed human membrane channel. Since both the N and C termini of AQP2 are trapped between the two crystal layers, their structure is sufficiently well ordered for crystallographic analysis. This information will help to understand the regulated shuttling of AQP2 to the apical membrane of the principal collecting duct cells.

Results and Discussion

Crystallization

Solubilized and highly purified N-terminally His-tagged AQP2 (HT-AQP2), previously shown to be functionally active and to exist as a tetramer,[32] was reconstituted into crystalline protein-lipid arrays. Initial crystallization trials were conducted in dialysis buttons at room temperature, while the most favorable conditions were determined using a temperature-controlled dialysis machine.[33] Table 1 summarizes the conditions tested and the optimal conditions found.

At pH 6.0, HT-AQP2 readily incorporated into proteoliposomes in the presence of *Escherichia coli* lipids or heart polar lipids, but not with dioleoylphosphatidylcholine (DOPC) or dimyristoylphosphatidylcholine (DMPC). Proteoliposomes of *E. coli* lipids clustered together more strongly than those containing heart polar lipids. At pH 5.0 only aggregates were found. When the pH was between 7.0 and 8.0 the vesicles were smaller, were mixed with tube-like structures and also had a propensity to aggregate. The protein concentration had a significant influence on the crystallinity of reconstituted HT-AQP2: Higher protein concentrations resulted in better-ordered protein-lipid arrays. The highest protein concentration used was 0.7 mg/ml for which the optimal lipid-to-protein ratio (LPR) was 0.5. Addition of Mg^{2+}, either in the form of $MgCl_2$ or as $MgSO_4$, dramatically improved the crystal quality. For reasons unknown, $MgCl_2$ seemed to work slightly better for *E. coli* lipids, whereas $MgSO_4$ gave better results for heart polar lipids. At concentrations above 5 mM, however, Mg^{2+} led to severe aggregation of the HT-AQP2 protein. It is not clear whether the His-tag at the protein's N terminus was responsible for this phenomenon, since similar aggregation previously observed for His-tagged AqpZ was also found after proteolytic elimination of the His-tag.[34] The concentration of NaCl had little influence on crystallinity, and was found to be optimal at 100 mM. Of the other additives tested, only histidine had a positive effect on crystal quality. At a concentration of 5 mM, it improved long-range crystallinity and reduced stacking of the 2D crystals.

However, the best 2D crystals were produced when a temperature-controlled dialysis machine was used for detergent removal. Instead of the round vesicle-like structures observed after using dialysis buttons (Figure 1(a)), large rectangular

Table 1. Summary of 2D-crystallization experiments

Parameter	Range tested	Optimal condition
Lipid type	DOPC, DMPC, *E. coli* lipids, Heart polar lipids	Heart polar lipids or *E. coli* lipids
Protein concentration	0.35–0.7 mg/ml	0.7
LPR	0.4–1.2	0.5
Buffer	Citric acid, Mes, K_2HPO_4/Na_2HPO_4, Tris	20 mM Mes
pH	5.0–8.0	6.0
NaCl	100–600 mM	100 mM
$MgCl_2$ or $MgSO_4$	0–20 mM	5 mM
DTT	0–10 mM	–
$NiCl_2$	0–10 mM	–
$CaCl_2$	0–10 mM	–
Histidine	0–50 mM	5 mM

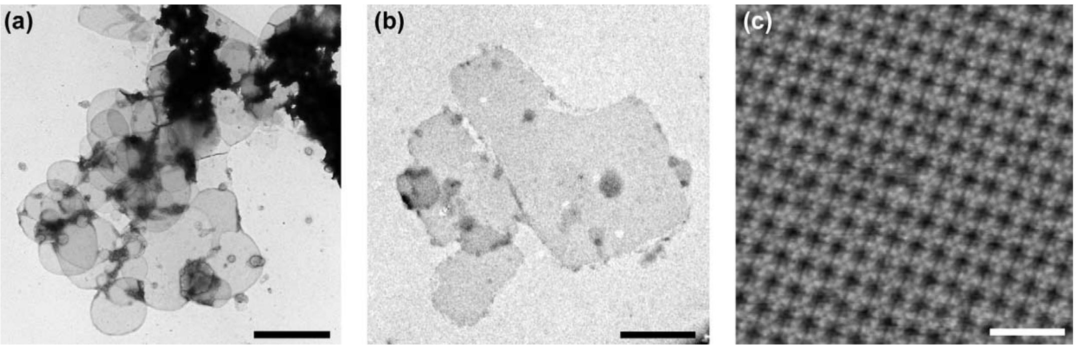


Figure 1. Electron microscopy of 2D AQP2 crystals. (a) Electron micrograph of a negatively stained sample of AQP2 2D crystals produced in dialysis buttons. Scale bar corresponds to 2 μm. The 2D crystals appear as vesicle-like structures, often clustered together and attached to aggregates. (b) Cryo-electron microscopic overview of AQP2 2D crystals obtained under optimized conditions in a dialysis machine. 2D crystals appear as large mono-crystalline rectangular protein-lipid arrays. Scale bar corresponds to 2 μm. (c) High-resolution AFM topograph of an AQP2 crystal sheet recorded at low force in an optimized imaging buffer. Scale bar corresponds to 150 Å.

protein-lipid 2D crystals were obtained (Figure 1(b)), whose surface topography recorded by AFM revealed mono-crystalline arrays of highly ordered tetramers (Figure 1(c)). Although the initial order of crystals prepared using heart polar lipid was slightly higher than that of crystals formed in the presence of *E. coli* lipids, these differences disappeared after a few weeks of storage at 4 °C. In fact, both heart polar lipid and *E. coli* lipid HT-AQP2 crystals improved in quality during this period of time, finally becoming indistinguishable in terms of unit cell dimensions, geometry and symmetry. This was rather unexpected, because lipids have often been observed to dictate the crystallization process in terms of order and morphology. Pertinent examples are Bacteriorhodopsin, which requires the native purple membrane lipids,[35] and the porin OmpF, which assembles into different lattice types depending on the nature of the lipid and the LPR used.[36] With AQP1, polymorphism depending on the LPR and the presence of Mg^{2+},[37] has been reported. The fact that HT-AQP2 formed identical crystals with two different lipid extracts suggests that co-purified endogenous lipids and/or protein–protein contacts have an important influence on the packing arrangement of the HT-AQP2 arrays, excluding extraneous lipids. This interpretation is compatible with the large lipid areas observed during AFM analyses of these crystals (see below).

Characterization of the AQP2 crystals by TEM

Micrographs of negatively stained 2D crystal preparations, optimized as described above, revealed rectangular mono-crystalline arrays of up to 30 μm side length (Figure 1(b)) that diffracted to better than $15\ \text{Å}^{-1}$. Such arrays were highly ordered as demonstrated by electron diffraction of trehalose-embedded HT-AQP2 crystals, which

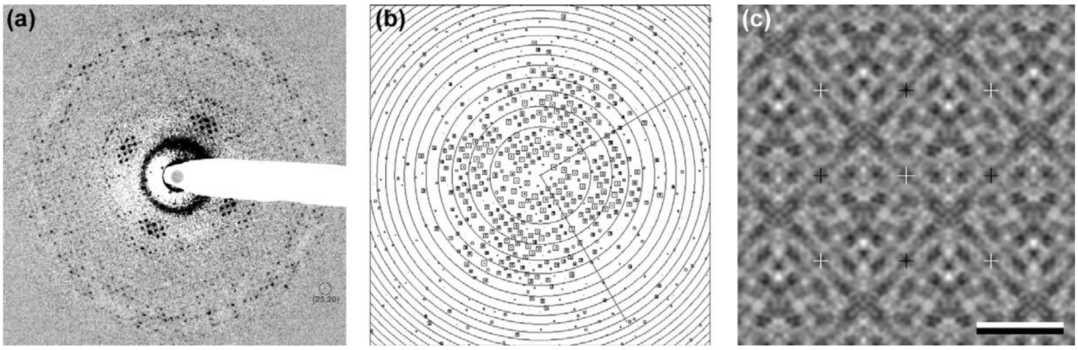


Figure 2. Cryo-electron microscopic analysis of 2D AQP2 crystals. (a) Typical electron diffraction pattern of an AQP2 crystal, revealing diffraction up to order (25,20), which corresponds to a resolution of $3.03\ \text{Å}^{-1}$. (b) IQ plot of a cryo electron micrograph of an AQP2 crystal. The IQ plot extends to $4.66\ \text{Å}^{-1}$. (c) Cryo projection map calculated by merging the phase information from ten well preserved AQP2 crystals and the amplitude information from 31 diffraction patterns, and imposing $p22_12$ symmetry. Scale bar corresponds to 50 Å.

showed orders up to (25,20) (Figure 2(a)), corresponding to a resolution of 3.03 Å^{-1}. Unexpectedly, however, the unit cell projection maps did not reveal the characteristic pattern of AQP tetramers (data not shown). Exploring possible crystallographic packing arrangements by ALLSPACE of the MRC suite, the best symmetry turned out to be $p22_12$ with a phase residual of 38° considering 460 reflections in total. The map shown in Figure 2(c) was calculated by taking the phases of ten projection maps up to a resolution of 4.9 Å (example given in Figure 2(b)), imposing $p22_12$ symmetry, and combining them with amplitudes from 31 electron diffraction patterns.

Mass-per-area determination (MPA)

Mass-per-area (MPA) measurements performed with the scanning transmission electron microscope (STEM[38]) yielded a histogram with several distinct peaks (Figure 3; low mass range only; excluding values from lipid areas). The peaks at 3.6(±0.2) kDa/nm^2 ($n=49$) and 6.7(±0.2) kDa/nm^2 ($n=335$) correspond to those observed for AQP0 single and double-layered sheets,[39] with MPAs 3.4(±0.3) kDa/nm^2 and 6.6(±0.4) kDa/nm^2, respectively, and for AqpZ single layer sheets,[34] MPA 3.2(±0.1) kDa/nm^2. Mainly as a result of a large glycane and the slightly denser packing, single-layered crystalline sheets of AQP1 have an MPA of 4.1(±0.3) kDa/nm^2.[40] Although double-layered 2D crystals were the most predominant form of the HT-AQP2 crystals, stacks thereof and of unfilled lipid bilayers were also found (data not shown).

Characterization of the AQP2 crystals by AFM

Atomic force topographs recorded from HT-AQP2 2D crystals adsorbed to atomically flat mica showed 2D crystals with a thickness of 118(±2) Å ($n=21$). Compared to single-layered 2D crystals of AQP1 (58(±4) Å[41,42]) and single-layered 2D crystals of the bacterial aquaporin AqpZ (57(±4) Å[43]) this showed that the HT-AQP2 crystals measured were double-layered. While a few single-layers were found during the STEM measurements, no single-layers were detected in the course of the AFM experiments. However, protein-free lipid areas having a thickness of about 4 nm were frequently observed. The same AQP2-surface was always exposed to the AFM stylus, as clearly evident from the high-resolution topographs recorded under optimized buffer conditions using low applied forces[44] (Figure 1(c)). Since no feature of these clean HT-AQP2 topographs had any similarity to the highly corrugated nature of His-tag-bearing AqpZ crystals,[43] the AQP2 surface characterized by AFM is most likely the extracellular one. All of the high-resolution topographs exhibit the same characteristics (averaged in Figure 4(a)). The 98 Å unit cell with $p4$ symmetry contains two non-crystallographically related but identical AQP2 tetramers, one rotated clockwise by 45.0(±1.8) degrees ($n=280$) around the tetramer's internal 4-fold axis, the other rotated clockwise by 53.7(±1.5) degrees ($n=247$), taking the monomer-monomer interface and lattice vectors for angle-determination. Interestingly, these two tetramers protruded out of the membrane with small, yet significant differences in height: 16.4(±1.7) Å ($n=280$) and 15.3(±1.9) Å ($n=247$), respectively.

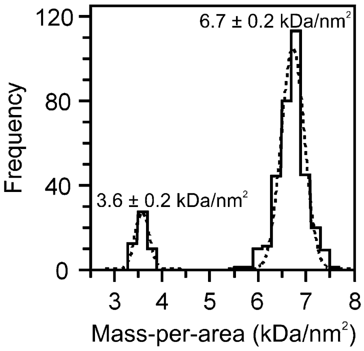


Figure 3. STEM mass/area measurements of 2D AQP2 crystals. Mass/area values binned in a histogram were fitted with Gaussian curves. The peak at 3.6(±0.2) kDa/nm^2 representing single sheets comprises 49 measurements and the peak resulting from double layers at 6.7(±0.2) kDa/nm^2 is from 335 measurements.

The HT-AQP2 topographs resemble those of the extracellular AQP0 surface.[45] The similarity is pronounced in that these proteins lack the prominent protrusions observed around the tetramer's 4-fold axis for AqpZ[43] and AQP1.[46] This is the result of the particularly short A-loop in both AQP2 and AQP0, which is longer in AqpZ and even longer in AQP1.[6] Taken together, these results strongly suggest that HT-AQP2 2D crystals only expose the extra-cellular side of HT-AQP2 to the medium, and that the cytoplasmic N and C termini of HT-AQP2 are trapped within the double-layered sandwich.

Layer separation

Electron diffraction patterns (Figure 2(a)) of double-layered HT-AQP2 2D crystals showed single lattices, rather than the epitaxial twinned lattices expected from double-layers rotated randomly with respect to each other. Thus, the double-layers were well aligned angularly, although possibly shifted. To determine the shift vector κ between the top and bottom layer in the double-layered HT-AQP2 2D crystals, a synthetic projection map of a single layer was generated based on AFM topographs of HT-AQP2 2D crystals (Figure 4(a)) and the AQP1 PDB model 1H6I (see Materials and Methods). This single layer model shown in Figure 4(b) was used for cross-correlation

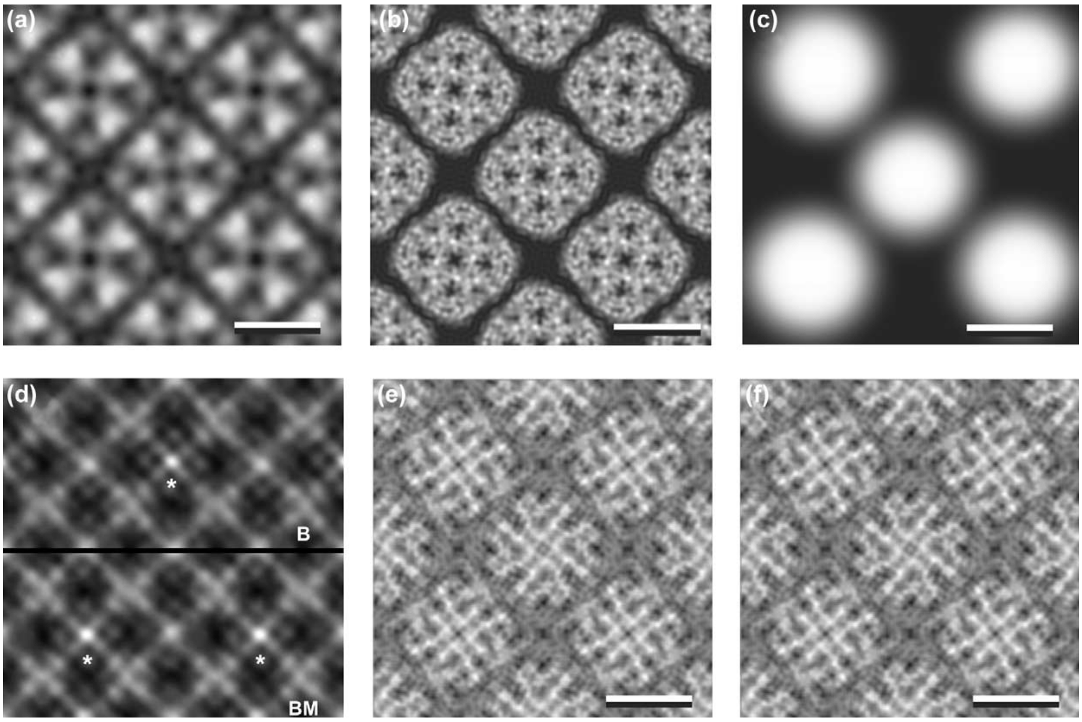


Figure 4. Iterative layer separation of 2D AQP2 crystals. (a) Fourier-filtered AFM topograph of an AQP2 crystal, revealing two unique tetramers within the unit cell, rotated by nine degrees with respect to each other. The AQP2 unit cell exhibits $p4$ symmetry and has a size of 98 Å. (b) Single-layered AQP2 model obtained by projecting the atomic structure of AQP1 along the 4-fold axis and taking the geometrical data from the AFM topograph. (c) Low-resolution model of an AQP2 single layer, as used for iterative layer separation. (d) Cross-correlation between an unsymmetrized AQP2 projection map and either the top layer model shown in (b), (cross-correlation function B), or the bottom layer model (i.e. mirrored top layer; cross-correlation function BM). The highest correlation peaks indicated by asterisks document the shift of half a unit cell between top and bottom layers. (e) Iterative layer separation result using the AQP2 model shown in (b), and (f) using the low-resolution model shown in (c). Results shown are after 1000 cycles. Both starting models converged to very similar single layer projections of AQP2 (differences <0.2%). For details see Materials and Methods. Scale bars correspond to 50 Å.

with the unsymmetrized AQP2 crystal projection map either directly (to determine the top layer position (Figure 4(d), B)) or after mirroring (to determine the bottom layer position (Figure 4(d), BM)). As indicated by the position of the highest cross-correlation peaks in Figure 4(d), the shift between the two layers of the AQP2 double-layered crystal corresponds to half a unit cell length, i.e. $\kappa=(0.5,0)$, resulting in an overall $p22_12$ symmetry for the double layer.

As demonstrated previously, stacked layers can be deconvoluted to produce single layer projection maps.[47] However, the deconvolution approach cannot be applied to double-layered crystals, whose layers are shifted with respect to each other by an integer fraction (i.e. $\kappa=1/n$, $1/m$; with n, m being integers). In this case systematic absences of diffraction orders implicate division by zero (see Materials and Methods). Because one would expect that for all highly ordered double-layered crystals the shift will be an integer fraction of the unit cell along one or both of the lattice vectors to warrant a precise, repetitive interaction between the layers, the previously proposed deconvolution method is not useful. Therefore, a novel, iterative real-space algorithm was developed that does not need division, but requires the single layer to have a plane-group symmetry higher than that of the double layer, which can be revealed by various techniques, e.g. by AFM (see Materials and Methods).

When the single layer model calculated from the AQP1 structure was used as the initial model (Figure 4(b)), the algorithm produced a projection map typical of single-layered aquaporin crystals (Figure 4(e)). The iteration process converged rapidly since the contribution of the shifted, second layer was reduced by 1/4 for each cycle, whereas that of the first layer centered about the 4-fold axis was not attenuated. The rotation angles of the two unique tetramers in this map, 45° and 54°, compare favorably to those determined by AFM. A single layer projection map that was virtually identical to the one calculated with the AQP1-based starting

Table 2. Electron crystallographic data

Plane group symmetry	$P222_1$
Unit cell	$a=b=98$ Å; $c=140$ Å (assumed)
	alpha=beta=gamma=90°
Number of processed images	363 (0°:69 15°:115 20°:6 30°:23 45°:150)
Number of merged phases	22,068
Resolution limit for merging	4.5 Å (in the membrane plane; *x*,*y*-direction)
	6.67 Å (perpendicular to the membrane plane; *z*-direction)
Phase residual (IQ-weighted)[a]	37.1° (Overall)
	26.9° (100–9.7 Å)
	33.2° (9.7–6.9 Å)
	67.8° (6.9–5.6 Å)
	85.7° (5.6–4.9 Å)
	86.5° (4.9–4.5 Å)
Completeness[b]	20% (resolution volume: 4.5 Å)
	60% (resolution volume 6.67 Å)

[a] Determined by AVRGAMPHS (MRC program suite).

[b] Determined by SFCHECK (CCP4), the missing cone is included in this volume.

model was obtained even when a coarse initial single layer model, e.g. circular blobs representing the HT-AQP2 tetramers (Figure 4(c)), was used (compare Figure 4(e) with (f)). Indeed, any coarse single layer starting model, such as the topograph acquired by AFM, produced a map similar to those shown in Figure 4(e) and (f), provided that the values of order (*h*, *k*) that vanish in the transform of the double layer projection map are also small in the starting model. Therefore, the algorithm developed resets such orders to 0 in each cycle of the iteration. This resulted in a robust and efficient layer separation method that is generally applicable, provided that the plane-group symmetry of the single layer and the shift vector are known.

The resolution of the reconstructed single layer projection map was determined by Fourier ring correlation analysis,[48] comparing the two non-crystallographically related tetramers. A resolution of 4.9 Å was obtained using the 0.5 criterion. Since the number of orders carrying structural information increases with the square of the resolution, but the number of systematic absences only linearly, the completeness of information is close to 97% at a resolution of 3 Å^{-1} and still about 95% at 5 Å^{-1} for the AQP2 lattices described here.

3D potential map

The information on the architecture of the double-layered crystals allowed projection maps of tilted samples to be interpreted, and the phase origins of projections recorded at lower tilt angles (10–20°) to be determined. An initial 3D map thus reconstructed made it possible to subsequently determine the phase origins of projections recorded at higher tilt angles (30–45°), and to merge the data set. In total the information was extracted from 363 images, of which 150 images were recorded at a tilt angle of 45° (Table 2). Image shift due to beam-induced specimen charging was a major problem when recording images at tilt angles above 30°.

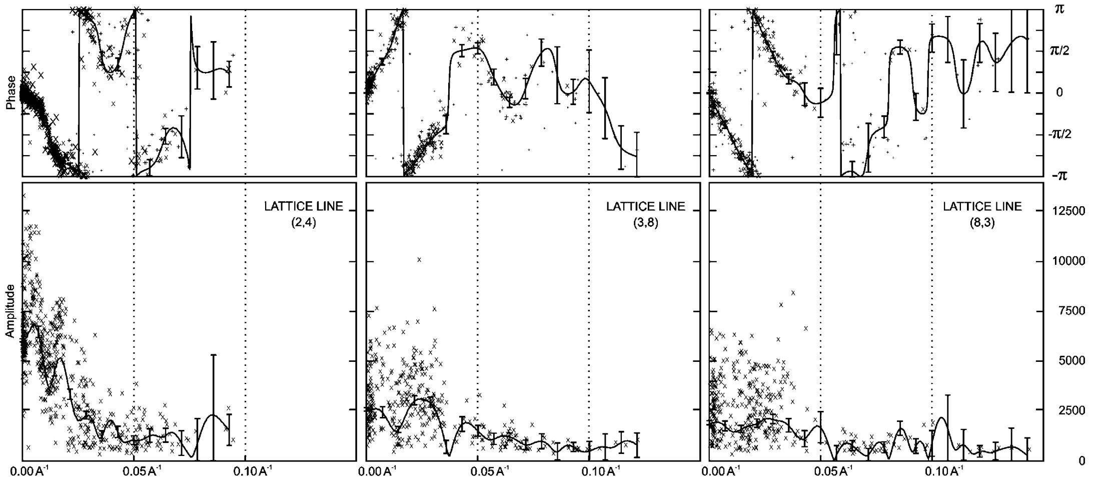


Figure 5. Lattice lines. Amplitude-phase pairs were extracted from 363 projection maps calculated from micrographs recorded over a tilt angle range of 0° to 45°.

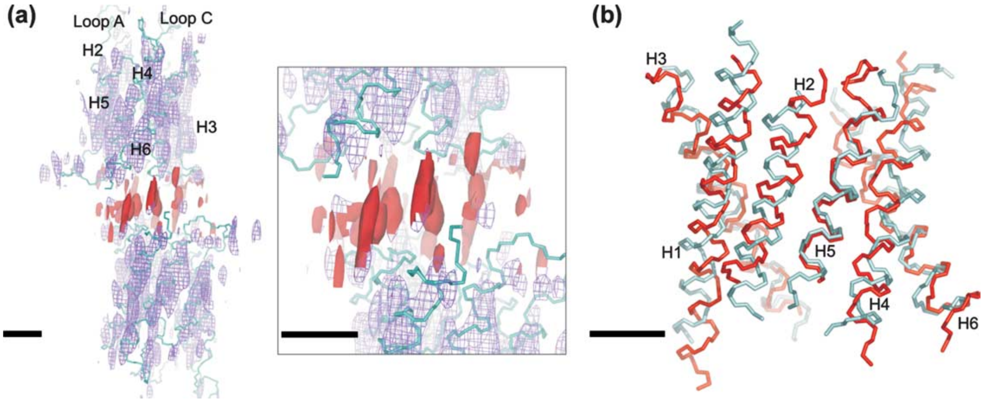


Figure 6. 3D potential map. (a) Two AQP2 monomers are shown in purple, one from the top layer, the other from the bottom layer. The fitted alpha-carbon backbone of AQP1 is displayed in cyan. The putative density of the termini is indicated in red. The inset provides a closer look at the termini section. (b) The helical fragments fitted to the map shown in (a) are colored in red and overlaid on the AQP1 helices in cyan. Scale bars represent 10 Å.

Attempts to overcome this difficulty included (i) the spot-scan mode,[49] (ii) evaporating a second carbon film onto glucose-embedded, air-dried samples,[50] and (iii) deposition of a second carbon film on the wet sample before freezing.[51] Each method has its own limitations. The first increases the difficulties in image processing, mainly in the lattice unbending procedure. The second method gave the most reproducible improvements, but can induce dehydration collapse if there is not sufficient support by the glucose. The third method is in theory the best, but it is experimentally the most difficult. Therefore, the quality of images recorded at 60° tilt was inferior to that of images acquired at 45°, explaining the lack of 60° projection data in our current data set.

Some of the lattice line data are shown in Figure 5. They demonstrate the quality of the phase information along z^* up to a resolution of 10 Å, while the amplitude data exhibit a significant scatter. The resolution range between 10 and 7 Å is sparsely populated, but still carries significant information. The 3D potential map calculated from this data set is displayed in Figure 6, showing two monomers, one from the upper layer and the other from the lower layer. The double layer has a thickness of 117 Å, compatible with the value acquired by AFM (118(±2) Å).

When the alpha-carbon backbone of AQP1 was fitted to the AQP2 3D map an intermediate layer of densities became visible, which could not be explained by any features of the AQP1 structures available. These structures do not provide information about the last 36 (hAQP1; 1H6I[9]) or 22 (bAQP1; 1J4N[11]) C-terminal residues. The major difference between the structure of AQP1 and that of AQP2 presented here is therefore most likely the result of contributions from the N and C termini, which appear to dictate the packing arrangement of the double-layered 2D crystals.

To further explore differences between AQP1 and AQP2, the program ROTTRANS[52] was used to fit helical segments to the 3D potential map of AQP2. Although the resolution along z^* did not suffice to identify the direction of the helices, the fitted segments allowed the tilt angles of transmembrane helices to be determined. The helices of AQP2 exhibit tilt angles with respect to the z-axis that are rather close to those of AQP1. However, the differences in angles between respective helices range from a few degrees up to more than ten degrees: H1 (2.0°); H2 (9.1°); H3 (5.1°); H4 (4.6°); H5 (10.8°) and H6 (6.0°). The largest differences are found in H2 and H5, which are the shortest of the six transmembrane helices.

Conclusion

The double-layered 2D crystals of HT-AQP2 described here are highly ordered and provides a solid basis to assess the atomic structure of this medically important aquaporin. Although it is challenging to establish the 3D structure of double-layered 2D crystals they appear to be better ordered to start with, and to be more stable than single layered crystals. Single layer HT-AQP2 projection maps calculated by a novel, iterative algorithm using different starting models exhibit all the features expected for an aquaporin projection map at 4.9 Å resolution. This documents the usefulness of the layer separation algorithm introduced for processing projection maps of double-layered 2D crystals. The overall features of the AQP2 monomer resemble those of AQP1. However, a striking new feature is the layer of densities sandwiched between the two single-layered crystal sheets, which we attribute to the N and C termini. Although the vertical resolution of this map (7 Å) does not suffice to resolve these possibly intertwined but highly ordered termini, electron diffraction at high tilt angles is expected to unravel their structure. The current study represents a first step towards this goal and shows the potential of the 2D crystals produced.

Materials and Methods

Expression and purification of AQP2

Human AQP2 was recombinantly expressed as an N-terminally His-tagged protein (HT-AQP2) in the baculovirus/insect cell expression system, and purified as described earlier[32] with some minor modifications. Briefly, Sf9 cells were grown to a density of 1.5×10^6 cells/ml in a 15 liter Bioreactor (Applikon) containing ten liters of Insect Xpress medium (BioWhittaker). Cells were infected with HT-AQP2 encoding baculovirus at a multiplicity of infection (MOI) of 0.05. Five days after infection, they were harvested by ten minutes centrifugation at 5000*g* and 4 °C, and homogenized in ice-cold 5 mM Tris–HCl (pH 8.0), 100 mM NaCl by 20 strokes at 500 rpm in Potter-Elvehjem tubes, followed by 20 strokes of douncing. An equivalent of 1×10^8 cells/ml was used throughout the entire stripping and solubilization procedure. Crude membranes were pelleted by 30 minutes centrifugation at 100,000*g* and 4 °C. This membrane pellet was homogenized in 5 mM Tris (pH 8.0), 1 mM EDTA, 4 M urea as above, and centrifuged for 45 minutes at 100,000*g* and 4 °C. The resulting pellet was homogenized in 20 mM NaOH (pH 12) and centrifuged for 90 minutes at 100,000*g* and 4 °C. This pellet was subjected to two rounds of homogenization in 5 mM Tris (pH 8.0), 100 mM NaCl and 30 minutes centrifugation at 100,000*g* and 4 °C, to restore the pH. The final stripped pellet was homogenized in solubilization buffer (20 mM Tris (pH 8.0), 300 mM NaCl, 1 mM L-histidine, 0.01% (w/v) NaN_3) containing 4% *n*-octyl-β-D-glucopyranoside (OG), by 20 strokes at 500 rpm in Potter-Elvehjem tubes, and solubilized by stirring gently for two hours at 4 °C. Solubilized proteins were separated from insoluble material by 60 minutes centrifugation at 100,000*g* and 4 °C, and diluted by the addition of an equal volume of solubilization buffer. Ni-NTA beads were added to the solubilized proteins (15 µl of Ni-NTA beads per equivalent of 1×10^8 cells) and gently stirred at 4°C for four hours. The Ni-NTA beads were then centrifuged for 15 minutes at 4000*g* and 4 °C, washed once with solubilization buffer containing 2% OG, and packed onto spin columns (Promega, A7651) by gravity flow. After 45 minutes incubation of the columns with elution buffer (solubilization buffer containing 2% OG and 100 mM L-histidine; 30 µl per equivalent of 1×10^8 cells), pure HT-AQP2 was eluted by two minutes centrifugation at 2000 rpm in an Eppendorf centrifuge, and directly used for crystallization.

Crystallization of AQP2

Lipid stocks (Avanti Polar Lipids) were solubilized in 2% OG at a concentration of 3 mg/ml and mixed with freshly prepared HT-AQP2 for crystallization trials. Initial trials were conducted in dialysis buttons with a volume of 60 µl, covered with a 10 kDa-cutoff dialysis membrane and dialyzed by submerging in 1 l flasks with the respective dialysis buffer. Final optimizations of the crystallization conditions were performed in 100 µl volumes in a temperature-controlled dialysis machine,[33] using the following profile: 12 hours at 20 °C, 24 hours linear ramp to 37 °C, 24 hours at 37 °C, 12 hours linear ramp to 20 °C. A summary of the conditions tested is given in Table 1.

Scanning transmission electron microscopy

For mass measurement, 7 µl aliquots of the HT-AQP2 samples were adsorbed for 45 s to glow discharged thin carbon films that spanned a thick fenestrated carbon layer covering 200-mesh/inch, gold-plated copper grids. The grids were blotted, washed on four drops of quartz bi-distilled water and freeze-dried overnight in the microscope at −80°C and 5×10^{-8} Torr. To calibrate the instrument, tobacco mosaic virus particles (kindly supplied by Dr R. Diaz-Avalos, Institute of Molecular Biophysics, Florida State University) were similarly adsorbed to a separate grid and air-dried.

Dark-field images were recorded from the unstained samples using a Vacuum Generators STEM HB-5 interfaced to a modular computer system (Tietz Video and Image Processing Systems GmbH, D-8035 Gauting). The accelerating voltage was 80 kV. A nominal magnification of 200,000× and recording doses in the range of 350 electrons/nm^2 were employed. Details of the instrument's calibration for mass measurement may be found in the publication by Müller *et al.*[38]

The 512×512 pixel digital images were evaluated using the program package IMPSYS as described.[38] Accordingly, sheet areas were defined by circular boxes and the total scattering of each calculated. The average background scattering of empty carbon support film adjacent to the sheets was subtracted and the MPA of the 2D crystals calculated. The resulting values were corrected for beam-induced mass-loss based on the behavior of 2D AqpZ crystals.[34] The mass data were displayed in histograms and fitted by Gauss curves.

Atomic force microscopy

A stock solution of 2D HT-AQP2 crystals (0.7 mg/ml protein) was diluted 30-fold in imaging buffer (20 mM Tris (pH 7.8), 150 mM KCl, 25 mM $MgCl_2$) and adsorbed for 20–30 minutes to freshly cleaved muscovite mica. After adsorption, the sample was gently washed with imaging buffer to remove membranes that were not firmly attached to the substrate. AFM experiments were performed using a Nanoscope III AFM (Digital Instruments, Veeco Metrology Group, Santa Barbara, CA, USA) equipped with a 150 µm J-Scanner, a fluid cell, and oxide-sharpened silicon nitride cantilevers of 100 µm length, with nominal spring constants of 0.09 N/m (Olympus Optical Co. and Digital Instruments, respectively). Topographs were acquired in contact mode at minimal loading forces (<100 pN).[44] Trace and retrace signals were recorded simultaneously at line frequencies ranging between 4.1 and 5.1 Hz.

Transmission electron microscopy

For imaging of negatively stained HT-AQP2 crystals, the protein sample was adsorbed for 5 s to glow-discharged, carbon film-coated copper grids and subsequently stained with 0.75% uranyl formate. Images were recorded on Kodak SO-163 film on a Hitachi H8000 TEM, at 100 kV and a nominal magnification of 50,000×.

Cryo-electron microscopy images and diffraction patterns were recorded from unstained samples, using a Philips CM200FEG TEM equipped with a Gatan cryo stage. Molybdenum grids (a kind gift from Dr Y. Fujiyoshi) coated with carbon films were prepared by the back-injection technique[53] using 2% (w/v) trehalose as embedding medium. Sample grids were frozen in liquid nitrogen and transferred to a Gatan cryo holder. Alternatively, samples were prepared in 2% (w/v) glucose and dried before cooling in the cryo holder. Diffraction patterns were recorded at 95 K, 200 kV, and a camera length of 1 m, using a Gatan UltraScan™ 2k×2k CCD camera. Images were recorded on Kodak SO-163 film at 200 kV and a nominal magnification of 50,000×, keeping the sample at 95 K. Attempts to overcome film charging included the spot scan mode,[49] evaporating a carbon layer onto the dried crystals,[50] or depositing a second carbon film onto the wet sample.[51]

Image processing

Electron micrographs were digitized with a Heidelberg Primescan D 7100 at 1 Å/pixel at the specimen level. Images and diffraction patterns recorded by cryo-EM were processed using the MRC software package.[54–56] Images were unbent three times, using Fourier-filtered versions of the respective images as a reference. Indexing of images was performed automatically using the SPIDER software package[57] to perform a peak search and a self-written program to determine the crystal lattice. The crystal symmetry was determined by the MRC program ALLSPACE. Electron diffraction patterns were corrected for background arising from inelastically scattered electrons and indexed using the MRC programs BACKAUTO and AUTOINDEX. Integration of the diffraction spots was carried out using the program PICKAUTO of the MRC package. A self-written program that calculates and applies scale factors to the diffraction data before combining the two datasets performed merging of diffraction amplitudes with image amplitudes and phases.

Layer separation

For the case of a double layer projection normal to the plane of the membrane the Fourier transform $G_d(h,k)$ can be written as:

$$G_d(h,k) = G_b(h,k) + G_t(h,k)\{\exp(-2\pi i\,\xi h)\} \qquad (1)$$

where $G_b(h,k)$ and $G_t(h,k)$ represent the transforms of the bottom and the top layer, h,k the indices of the diffraction order, and ξ is the shift along the x-axis in fractions of the unit cell length. If layers are stacked unidirectionally, the transform of the single layer is obtained by deconvolution:

$$G_b(h,k) = G_t(h,k) = G_d(h,k)/(1 + \{\exp(-2\pi i\,\xi h)\}) \qquad (2)$$

As a result of systematic absences for orders ($\pm 2n+1$, k), deconvolution cannot be applied when the shift $\xi = 0.5$ or generally $1/n$ with n being an integer. For double layers that are stacked face-to-face by rotation of one layer by π around the y-axis, the transforms of the single layer projections are related by:

$$G_b(h,k) = G_t(-h,k) \qquad (3)$$

For single layers lacking mirror symmetry, identities $G_b(h,k) = G_t(h,k)$ are found for orders ($\pm n$,0) if the single layers exhibit $p2$ symmetry, and additionally for orders $(0, \pm n)$ and $(\pm n, \pm n)$ if the single layers have $p4$ symmetry. Systematic absences thus occur for orders $h,k = \pm 2n+1$, 0 ($p2$), and additionally for diagonal orders $h = k = \pm 2n+1$ ($p4$) if double layers are shifted with respect to each other by half a unit cell along the x-axis, again preventing the application of the separation specified by equation (2). Therefore, a real-space iteration method was used to generate a single layer projection map from the double-layered HT-AQP2 crystals. To determine the shift between the two layers, the *iplt* image processing software tools[58] were used to cross-correlate the double layer projection map with a single layer model and its mirror image. The shift vector κ was obtained from the position of the cross-correlation maxima. The single layer model was constructed using a combination of geometry information from AFM topographs of HT-AQP2 crystals and the calculated electron density map of tetramers built from the AQP1 PDB file 1H6I.[9] The software SEMPER[59] was used to determine the position and rotational angle of the individual tetramers in the HT-AQP2 unit cell derived from AFM topographs, and programs from the GROMACS package† were used to build the AQP1 tetramers. The projection map of the single layer model was generated with the CCP4 suite‡.[60]

A high-resolution single layer projection map was then obtained by a real-space iteration algorithm, implemented as an *iplt* script. This iteration was carried out with different starting models to test the robustness of the method: (i) with the AQP1 single layer model (described above), (ii) with an artificial model (simple blobs representing the AQP2 tetramers), (iii) with random noise, and (iv) with the topograph acquired by AFM. In each case, the reference was scaled such that the half peak width of its intensity distribution equaled one times that of the double layer. Subtraction of the single layer model from the double layer yielded a new single layer model. This new single layer model, now carrying more information from the AQP2 projection map, was mirrored, shifted by κ and 4-fold symmetrized. To improve the robustness of the method, reflections that were systematically absent in the Fourier transform of the double layer were also set to zero in the single layer model. The new model was averaged with the model from the previous cycle, scaled and subtracted from the double layer. This procedure was repeated until the result converged.

Calculation of the 3D potential map

The tilt geometry of the micrographs was determined by measuring the defocus by CTF-fitting in 7×7 sub-areas of the image and then fitting a least squares plane through the 49 points. For the projections recorded at low tilt angle the phase origins were first determined by eye. Then the images were merged and scaled using the program ORIGTILT and lattice line curves were fitted with LATLINE, both of the MRC program suite. The obtained dataset then served as reference for finding the phase origin and scaling factor of the projections taken at higher tilt angles. In a cyclical refinement procedure the tilt geometry, beam tilt, phase origin and scaling factor were improved to obtain a more accurate potential map. At the start of each cycle, every individual image was compared with the fitted dataset using ORIGTILT to determine new values for the image parameters mentioned above. For the next cycle, all images were merged using the new

---

† http://www.gromacs.org
‡ http://www.ccp4.ac.uk

image parameters and the LATLINE program was used to generate a new reference data set.

Helix-fitting

Helical fragments were fitted to the 3D potential map of AQP2 as described.[52] The program ROTTRANS (obtainable free†) was used for full rotation and translation searches through the AQP2 3D map. A set of 38 α-helical fragments was extracted from the Protein Data Bank, mutated to poly-valine helices, and taken as probe helices for ROTTRANS. Positions and tilt angles of the six transmembrane helices were then derived from the fitted helical segments and compared to the same parameters calculated for AQP1.

Acknowledgements

We thank Dr Y. Fujiyoshi for a kind gift of Molybdenum grids, Dr R. Henderson for providing a copy of the CONVDECONV program, Dr R. Diaz-Avalos for the tobacco mosaic virus particles and Vesna Olivieri for the STEM microscopy. We are thankful to Dr Peter Dean for his strong support in the early phase of the project and to Dr Wim de Grip for his help with the large-scale expression of AQP2. This work was supported by the Swiss National Science Foundation (grant number NF31-59415.99) to A.E., the M.E. Müller foundation of Switzerland, the National Center of Competence in Research (NCCR) for Structural Biology, and the EC-CONTRACT QLK3-1999-00778.

References

1. Agre, P. & Kozono, D. (2003). Aquaporin water channels: molecular mechanisms for human diseases. *FEBS Letters*, **555**, 72–78.
2. Agre, P., King, L. S., Yasui, M., Guggino, W. B., Ottersen, O. P., Fujiyoshi, Y. *et al.* (2002). Aquaporin water channels-from atomic structure to clinical medicine. *J. Physiol.* **542**, 3–16.
3. King, L. S., Yasui, M. & Agre, P. (2000). Aquaporins in health and disease. *Mol. Med. Today*, **6**, 60–65.
4. Deen, P. M. T. & van Os, C. H. (1998). Epithelial aquaporins. *Curr. Opin. Cell Biol.* **10**, 435–442.
5. Engel, A., Fujiyoshi, Y. & Agre, P. (2000). The importance of aquaporin water channel protein structures. *EMBO J.* **19**, 800–806.
6. Heymann, J. B. & Engel, A. (2000). Structural clues in the sequences of the aquaporins. *J. Mol. Biol.* **295**, 1039–1053.
7. Jung, J., Preston, G., Smith, B., Guggino, W. & Agre, P. (1994). Molecular structure of the water channel through aquaporin CHIP. The hourglass model. *J. Biol. Chem.* **269**, 14648–14654.
8. Murata, K., Mitsuoka, K., Hirai, T., Walz, T., Agre, P., Heymann, J. B. *et al.* (2000). Structural determinants of water permeation through aquaporin-1. *Nature*, **407**, 599–605.
9. de Groot, B. L., Engel, A. & Grubmuller, H. (2001). A refined structure of human aquaporin-1. *FEBS Letters*, **504**, 206–211.
10. Ren, G., Reddy, V. S., Cheng, A., Melnyk, P. & Mitra, A. K. (2001). Visualization of a water-selective pore by electron crystallography in vitreous ice. *Proc. Natl Acad. Sci. USA*, **98**, 1398–1403.
11. Sui, H., Han, B. G., Lee, J. K., Walian, P. & Jap, B. K. (2001). Structural basis of water-specific transport through the AQP1 water channel. *Nature*, **414**, 872–878.
12. Gonen, T., Sliz, P., Kistler, J., Cheng, Y. & Walz, T. (2004). Aquaporin-0 membrane junctions reveal the structure of a closed water pore. *Nature*, **429**, 193–197.
13. Harries, W. E., Akhavan, D., Miercke, L. J., Khademi, S. & Stroud, R. M. (2004). The channel architecture of aquaporin 0 at a 2.2-A resolution. *Proc. Natl Acad. Sci. USA*, **101**, 14045–14050.
14. Fu, D., Libson, A., Miercke, L. J., Weitzman, C., Nollert, P., Krucinski, J. & Stroud, R. M. (2000). Structure of a glycerol-conducting channel and the basis for its selectivity. *Science*, **290**, 481–486.
15. Tajkhorshid, E., Nollert, P., Jensen, M. O., Miercke, L. J., O'Connell, J., Stroud, R. M. & Schulten, K. (2002). Control of the selectivity of the aquaporin water channel family by global orientational tuning. *Science*, **296**, 525–530.
16. Savage, D. F., Egea, P. F., Robles-Colmenares, Y., Iii, J. D. & Stroud, R. M. (2003). Architecture and selectivity in aquaporins: 2.5 Å X-ray structure of aquaporin Z. *PLoS Biol.* **1**, E72.
17. Nielsen, S., Frokiaer, J. & Knepper, M. A. (1998). Renal aquaporins: key roles in water balance and water balance disorders. *Curr. Opin. Nephrol. Hypertens.* **7**, 509–516.
18. Knepper, M. A., Verbalis, J. G. & Nielsen, S. (1997). Role of aquaporins in water balance disorders. *Curr. Opin. Nephrol. Hypertens.* **6**, 367–371.
19. Deen, P. M. T. & Knoers, N. V. A. M. (1998). Physiology and pathophysiology of the aquaporin-2 water channel. *Curr. Opin. Nephrol. Hypertens.* **7**, 37–42.
20. van Os, C. H. & Deen, P. M. T. (1998). Role of aquaporins in renal water handling: physiology and pathophysiology. *Nephrol. Dial. Transplant.* **13**, 1645–1651.
21. Mulders, S. M., Bichet, D. G., Rijss, J. P., Kamsteeg, E. J., Arthus, M. F., Lonergan, M. *et al.* (1998). An aquaporin-2 water channel mutant which causes autosomal dominant nephrogenic diabetes insipidus is retained in the Golgi complex. *J. Clin. Invest.* **102**, 57–66.
22. Deen, P. M. T., Verdijk, M. A. J., Knoers, N. V. A. M., Wieringa, B., Monnens, L. A. H., van Os, C. H. & van Oost, B. A. (1994). Requirement of human renal water channel aquaporin-2 for vasopressin- dependent concentration of urine. *Science*, **264**, 92–95.
23. Nielsen, S., Kwon, T. H., Christensen, B. M., Promeneur, D., Frokiaer, J. & Marples, D. (1999). Physiology and pathophysiology of renal aquaporins. *J. Am. Soc. Nephrol.* **10**, 647–663.
24. Xu, L., Poole, D. C. & Musch, T. I. (1998). Effect of heart failure on muscle capillary geometry: implications for 02 exchange. *Med. Sci. Sports Exerc.* **30**, 1230–1237.
25. Nielsen, S., Terris, J., Andersen, D., Ecelbarger, C.,

---

† http://www.mpibpc.mpg.de/groups/de_groot/bgroot/maptools.html

Frokiaer, J., Jonassen, T. *et al*. (1997). Congestive heart failure in rats is associated with increased expression and targeting of aquaporin-2 water channel in collecting duct. *Proc. Natl Acad. Sci. USA*, **94**, 5450–5455.

26. Schrier, R. W., Fassett, R. G., Ohara, M. & Martin, P. Y. (1998). Vasopressin release, water channels, and vasopressin antagonism in cardiac failure, cirrhosis, and pregnancy. *Proc. Assoc. Am. Physicians.* **110**, 407–411.
27. Jonassen, T. E., Nielsen, S., Christensen, S. & Petersen, J. S. (1998). Decreased vasopressin-mediated renal water reabsorption in rats with compensated liver cirrhosis. *Am. J. Physiol.* **275**, F216–F225.
28. Asahina, Y., Izumi, N., Enomoto, N., Sasaki, S., Fushimi, K., Marumo, F. & Sato, C. (1995). Increased gene expression of water channel in cirrhotic rat kidneys. *Hepatology*, **21**, 169–173.
29. Fernandez-Llama, P., Turner, R., Dibona, G. & Knepper, M. A. (1999). Renal expression of aquaporins in liver cirrhosis induced by chronic common bile duct ligation in rats. *J. Am. Soc. Nephrol.* **10**, 1950–1957.
30. Pearson, J. F. (1992). Fluid balance in severe preeclampsia. *Br. J. Hosp. Med.* **48**, 47–51.
31. Schrier, R. W., Ohara, M., Rogachev, B., Xu, L. & Knotek, M. (1998). Aquaporin-2 water channels and vasopressin antagonists in edematous disorders. *Mol. Genet. Metab.* **65**, 255–263.
32. Werten, P. J. L., Hasler, L., Koenderink, J. B., Klaassen, C. H. W., de Grip, W. J., Engel, A. & Deen, P. M. T. (2001). Large-scale purification of functional recombinant human aquaporin-2. *FEBS Letters*, **504**, 200–205.
33. Jap, B. K., Zulauf, M., Scheybani, T., Hefti, A., Baumeister, W., Aebi, U. & Engel, A. (1992). 2D crystallization: from art to science. *Ultramicroscopy*, **46**, 45–84.
34. Ringler, P., Borgnia, M. J., Stahlberg, H., Maloney, P. C., Agre, P. & Engel, A. (1999). Structure of the water channel AqpZ from *Escherichia coli* revealed by electron crystallography. *J. Mol. Biol.* **291**, 1181–1190.
35. Mitra, A. K., Miercke, L. J., Turner, G. J., Shand, R. F., Betlach, M. C. & Stroud, R. M. (1993). Two-dimensional crystallization of *Escherichia coli*-expressed bacteriorhodopsin and its D96N variant: high resolution structural studies in projection. *Biophys. J.* **65**, 1295–1306.
36. Engel, A., Hoenger, A., Hefti, A., Henn, C., Ford, R. C., Kistler, J. & Zulauf, M. (1992). Assembly of 2-D membrane protein crystals: dynamics, crystal order, and fidelity of structure analysis by electron microscopy. *J. Struct. Biol.* **109**, 219–234.
37. Ren, G., Cheng, A., Melnyk, P. & Mitra, A. K. (2000). Polymorphism in the packing of aquaporin-1 tetramers in 2-D crystals. *J. Struct. Biol.* **130**, 45–53.
38. Müller, S. A., Goldie, K. N., Burki, R., Haring, R. & Engel, A. (1992). Factors influencing the precision of quantitative scanning transmission electron microscopy. *Ultramicroscopy*, **46**, 317–334.
39. Hasler, L., Walz, T., Tittmann, P., Gross, H., Kistler, J. & Engel, A. (1998). Purified lens major intrinsic protein (MIP) forms highly ordered tetragonal two-dimensional arrays by reconstitution. *J. Mol. Biol.* **279**, 855–864.
40. Walz, T., Smith, B. L., Zeidel, M. L., Engel, A. & Agre, P. (1994). Biologically active 2-dimensional crystals of aquaporin CHIP. *J. Biol. Chem.* **269**, 1583–1586.
41. Walz, T., Tittmann, P., Fuchs, K. H., Muller, D. J., Smith, B. L., Agre, P. *et al*. (1996). Surface topographies at subnanometer-resolution reveal asymmetry and sidedness of aquaporin-1. *J. Mol. Biol.* **264**, 907–918.
42. Müller, D. J. & Engel, A. (1997). The height of biomolecules measured with the atomic force microscope depends on electrostatic interactions. *Biophys. J.* **73**, 1633–1644.
43. Scheuring, S., Ringler, P., Borgnia, M., Stahlberg, H., Müller, D. J., Agre, P. & Engel, A. (1999). High resolution AFM topographs of the *Escherichia coli* water channel aquaporin Z. *EMBO J.* **18**, 4981–4987.
44. Müller, D. J., Fotiadis, D., Scheuring, S., Müller, S. A. & Engel, A. (1999). Electrostatically balanced subnanometer imaging of biological specimens by atomic force microscope. *Biophys. J.* **76**, 1101–1111.
45. Fotiadis, D., Hasler, L., Müller, D. J., Stahlberg, H., Kistler, J. & Engel, A. (2000). Surface tongue-and-groove contours on lens MIP facilitate cell-to-cell adherence. *J. Mol. Biol.* **300**, 779–789.
46. Fotiadis, D., Suda, K., Tittmann, P., Jeno, P., Philippsen, A., Muller, D. J. *et al*. (2002). Identification and structure of a putative Ca2++-binding domain at the C terminus of AQP1. *J. Mol. Biol.* **318**, 1381–1394.
47. Kunji, E. R., Spudich, E. N., Grisshammer, R., Henderson, R. & Spudich, J. L. (2001). Electron crystallographic analysis of two-dimensional crystals of sensory rhodopsin II: a 6.9 Å projection structure. *J. Mol. Biol.* **308**, 279–293.
48. Saxton, W. O. & Baumeister, W. (1982). The correlation averaging of a regularly arranged bacterial cell envelope protein. *J. Microsc.* **127**, 127–138.
49. Downing, K. H. (1991). Spot-scan imaging in transmission electron microscopy. *Science*, **251**, 53–59.
50. Brink, J., Gross, H., Tittmann, P., Sherman, M. B. & Chiu, W. (1998). Reduction of charging in protein electron cryomicroscopy. *J. Microsc.* **191**, 67–73.
51. Gyobu, N., Tani, K., Hiroaki, Y., Kamegawa, A., Mitsuoka, K. & Fujiyoshi, Y. (2004). Improved specimen preparation for cryo-electron microscopy using a symmetric carbon sandwich technique. *J. Struct. Biol.* **146**, 325–333.
52. de Groot, B. L., Heymann, J. B., Engel, A., Mitsuoka, K., Fujiyoshi, Y. & Grubmuller, H. (2000). The fold of human aquaporin 1. *J. Mol. Biol.* **300**, 987–994.
53. Hirai, T., Murata, K., Mitsuoka, K., Kimura, Y. & Fujiyoshi, Y. (1999). Trehalose embedding technique for high-resolution electron crystallography: application to structural study on bacteriorhodopsin. *J. Electron Microsc. (Tokyo)*, **48**, 653–658.
54. Henderson, R., Baldwin, J. M., Ceska, T. A., Zemlin, F., Beckmann, E. & Downing, K. H. (1990). Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *J. Mol. Biol.* **213**, 899–929.
55. Henderson, R., Baldwin, J. M., Downing, K. H., Lepault, J. & Zemlin, F. (1986). Structure of purple membrane from halobacterium halobium: recording measurement and evaluation of electron micrographs at 3.5 Å resolution. *Ultramicroscopy*, **19**, 147–178.
56. Baldwin, J. M. & Henderson, R. (1984). Measurement and evaluation of electron diffraction patterns from two-dimensional crystals. *Ultramicroscopy*, **14**, 319–336.
57. Frank, J., Shimkin, B. & Dowse, H. (1981). SPIDER-a modular software system for electron image processing. *Ultramicroscopy*, **6**, 343–358.
58. Philippsen, A., Schenk, A. D., Stahlberg, H. & Engel,

A. (2003). Iplt-image processing library and toolkit for the electron microscopy community. *J. Struct. Biol.* **144**, 4–12.
59. Saxton, W. O. (1996). Semper: distortion compensation, selective averaging, 3-D reconstruction, and transfer function correction in a highly programmable system. *J. Struct. Biol.* **116**, 230–236.
60. Collaborative Computational Project, N. (1994). The CCP4 Suite: Programs for Protein Crystallography. *Acta Crystallog. sect. D*, **50**, 760–763.

Edited by Baumeister

(Received 24 August 2004; received in revised form 2 April 2005; accepted 15 April 2005)

2.3 The 5 Å Structure of Heterologously Expressed Plant Aquaporin SoPIP2;1

Contribution

This work published in 2005 in the Journal of Molecular Biology volume 350, pages 611-616, is part of the PhD thesis project of Wanda Kukulski, the main author of the publication. The purification and crystallization was solely achieved by her.

My contribution to this project consisted in improvement of the sample preparation (see 4.1) to allow recording of highly tilted images, recording the 60° tilted images included in the publication and coding the programs and scripts specifically used for the SoPIP2;1 project (see 3). The image processing of the initial tilted and untilted images, the recording of the first diffraction patterns, the determination of the symmetry and categorization of the imaged crystals into the two different classes with 65 Å and 92 Å unit cell, the first 3D reconstruction, the basic refinement of the 3D reconstruction and fitting of a first pdb model of AQP1 into the 3D density were done by me and Wanda Kukulski in a collaborative fashion, in the context of demonstrating the usage of the local image processing framework. Recording of most of the images and diffraction patterns, processing of later recorded images and further refinement of the 3D reconstruction were carried out by Wanda Kukulski herself. Subsequent fits of pdb structures into the 3D density were carried out by Bert de Groot.

2.3.1 Summary

SoPIP2;1 is an aquaporin present in the leaf plasma membrane of *Spinacia oleracea* (spinach). During crystallization SoPIP2;1 the most commonly observed double layered crystals exhibited a unit cell of 65 Å and p4 symmetry. In rare cases, crystals where found were every second SoPIP2;1 tetramer was slightly rotated, giving rise to a unit cell of 92 Å with $p22_12$ symmetry similar to AQP2. Sometimes, single layered crystals were found, which were clearly less well ordered than the double layered ones. The structure determination concentrated on the double layered crystals exhibiting a p4 symmetry because they were found to be most abundant and easier to process than the $p22_12$ crystal symmetry.

2.3.2 Addendum

Very recently the structure of the SoPIP2;1 water channel was determined in its closed conformation at 2.1 Å and in its open conformation at 3.9 Å.[42] The structure of the closed conformation shows that the loop D caps the channel from the cytoplasm and thereby occludes the pore. By displacement of loop D up to 16 Å, a hydrophobic gate, blocking the channel entrance from the cytoplasm, is opened. These results reveal a molecular gating mechanism, which appears conserved throughout all plant plasma membrane aquaporins.

doi:10.1016/j.jmb.2005.05.001 *J. Mol. Biol.* (2005) **350**, 611–616

JMB

Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

COMMUNICATION

The 5 Å Structure of Heterologously Expressed Plant Aquaporin SoPIP2;1

W. Kukulski[1], A. D. Schenk[1], U. Johanson[2], T. Braun[1], B. L. de Groot[3]
D. Fotiadis[1], P. Kjellbom[2] and A. Engel[1]*

[1]*Maurice E. Müller Institute for Microscopy, Biozentrum University of Basel, CH-4056 Basel, Switzerland*

[2]*Department of Plant Biochemistry, Lund University S-221 00 Lund, Sweden*

[3]*Computational Biomolecular Dynamics Group Max-Planck-Institute for Biophysical Chemistry, Am Fassberg 11, D-37077 Göttingen Germany*

*Corresponding author

SoPIP2;1 is one of the major integral proteins in spinach leaf plasma membranes. In the *Xenopus* oocyte expression system its water channel activity is regulated by phosphorylation at the C terminus and in the first cytosolic loop. To assess its structure, SoPIP2;1 was heterologously expressed in *Pichia pastoris* as a His-tagged protein and in the non-tagged form. Both forms were reconstituted into 2D crystals in the presence of lipids. Tubular crystals and double-layered crystalline sheets of non-tagged SoPIP2;1 were observed and analyzed by cryo-electron microscopy. Crystalline sheets were highly ordered and diffracted electrons to a resolution of 2.96 Å. High-resolution projection maps of tilted specimens provided a 3D structure at 5 Å resolution. Superposition of the SoPIP2;1 potential map with the atomic model of AQP1 demonstrates the generally well conserved overall structure of water channels. Differences concerning the extracellular loop A explain the particular crystal contacts between oppositely oriented membrane sheets of SoPIP2;1 2D crystals, and may have a function in rapid volume changes observed in stomatal guard cells or mesophyll protoplasts. This crystal packing arrangement provides access to the phosphorylated C terminus as well as the loop B phosphorylation site for studies of channel gating.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: aquaporin; electron diffraction; electron microscopy; three-dimensional structure; two-dimensional crystals

Cytosolic osmoregulation is maintained at the single-cell level and whole-plant water homeostasis is an essential function for every plant. It is accomplished by a complex and only rudimentarily understood system. While long-distance water transport is fulfilled by the vascular tissue, the cells themselves provide a rapid short-distance water flow through membranes. Proteins of the aquaporin family are key components in cellular water homeostasis, and they account for a significant fraction of the total amount of integral membrane proteins of plant plasma membranes.[1] Their importance is demonstrated also by the fact that in *Arabidopsis* 35 genes encode expressed aquaporin-like proteins and around one-third of these are located at the plasma membrane.[2]

At the cellular level, the maintenance of the water balance is an interplay between plasma membrane and tonoplast aquaporins (plasma membrane intrinsic proteins (PIPs) and tonoplast intrinsic proteins (TIPs), respectively). The PIPs are subdivided into two groups, the PIP1 and PIP2 isoforms. The latter have a longer C-terminal region and, when expressed in *Xenopus* oocytes, show a higher water transport activity than the PIP1 isoforms.[3]

SoPIP2;1, in previous nomenclature called PM28A, is a PIP2 isoform in *Spinacia oleracea* (spinach) leaf plasma membranes, where aquaporins constitute ∼20% of all integral membrane proteins.[4] In its C-terminal region, a serine residue was found to be phosphorylated *in vivo*, in response to increasing apoplastic water potential.[4] Results generated by using the *Xenopus* oocyte expression system in combination with site-directed

---

Abbreviations used: AQP, aquaporin; PIP, plasma membrane intrinsic protein; TIP, tonoplast intrinsic protein.

E-mail address of the corresponding author: andreas.engel@unibas.ch

0022-2836/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.

mutagenesis suggested that phosphorylation at Ser274 in the C-terminal region regulates the water channel activity of SoPIP2;1.[5] Furthermore, another serine residue, Ser115, in a consensus phosphorylation site in the first cytosolic loop, i.e. loop B, was identified as an additional putative regulatory phosphorylation site.[5] A hypothetical model has been presented in which an osmosensor senses the difference and triggers the increase of the intracellular concentration of Ca^{2+} when the apoplastic water potential is higher than the water potential in the cell.[5,6] This leads to the activation of a plasma membrane-associated Ca^{2+}-dependent protein kinase, which phosphorylates SoPIP2;1 at the Ser274 residue and thus facilitates water influx. At low apoplastic water potential, the channel is dephosphorylated and water flow through the pore is restricted. The consensus phosphorylation site at Ser274 in the C-terminal region is conserved in all PIP2 isoforms, independent of species, but not in the PIP1 isoforms, which have a shorter C-terminal region. The consensus phosphorylation site in the loop B at Ser115 is conserved in all PIPs, i.e. in all PIP1 as well as PIP2 isoforms.[2]

Several aquaporin structures have been solved in the past few years; human AQP1,[7,8] AQP0 from sheep[9] and bovine[10] eye lens, as well as two bacterial members, GlpF[11] and AqpZ.[12] Together with molecular dynamics simulations,[13–17] these structures gave insight into function and selectivity of water channels. In the case of AQP0, clues about its specific pH-dependent regulation could be derived,[9] and the structures of the open[10] and closed[9] AQP0 water channel are now available. The only plant aquaporin for which structural data are available is a projection map of the vacuolar membrane aquaporin α-TIP determined by electron crystallography.[18] Thus, the density map presented here is the first 3D plant aquaporin structure. Furthermore, we report here, to our knowledge, the first structure of a heterologously expressed aquaporin in any eukaryotic species.

SoPIP2;1 was expressed both as His-tagged and as non-tagged protein in the methylotrophic yeast *Pichia pastoris*. These clones both express SoPIP2;1 variants at similarly high levels. When reconstituted into proteoliposomes and exposed to an osmotic gradient, recombinant SoPIP2;1 shows efficient water channel activity.[19]

Both His-tagged and non-tagged SoPIP2;1 could be reconstituted into several crystal forms. Although more different forms were obtained with the His-tagged protein (data not shown), better-ordered crystals resulted from non-tagged SoPIP2;1. We were interested in the native SoPIP2;1 in the spinach plasma membrane, and so we concentrated on the structural analysis of crystals reconstituted in the presence of lipids and the non-tagged protein. Non-tagged SoPIP2;1 crystallized in two forms. Tubular vesicles exhibited a specific surface texture, resulting from up-down oriented tetramers that are packed into alternating rows (data not shown). Image processing analysis revealed these crystals to be anisotropically ordered, and thus not suitable for high-resolution structural analysis. The second crystal type concerns membrane sheets that are mostly double-layered, exhibit $p4$ symmetry, and have lattice constants of $a=b=65$ Å. In contrast to the coaxially packed double-layered AQP0 crystals,[9,20] the two crystalline layers of SoPIP2;1 are shifted against each other by exactly half a unit cell in the x and y direction, and are thus packed in precise register as well. The crystallographic unit cell comprises a tetramer from one layer and four monomers from four neighboring tetramers in the opposing layer. Occasionally recorded single layers are generally less well ordered, suggesting that a stabilizing crystal contact exists between the two layers.

The crystal quality was assessed by electron diffraction at low electron dose (<5 é/Å^2). Strong diffraction spots could be observed to a resolution of 2.96 Å on untilted crystals and to 3.7 Å at a tilt angle of 60° (Figure 1). For structure determination

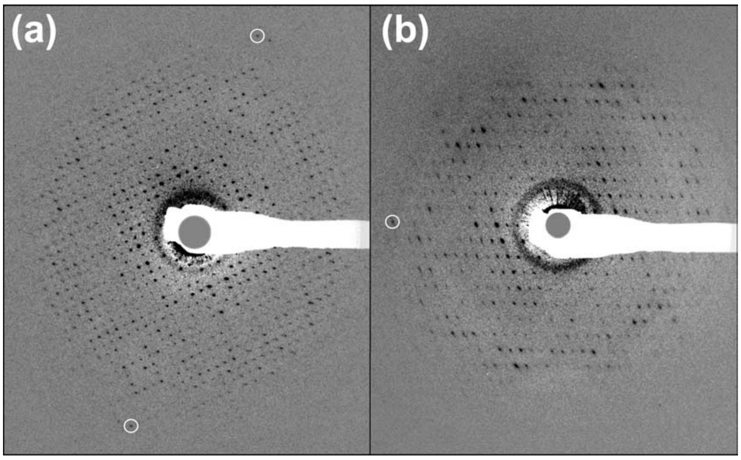


Figure 1. Electron diffraction of double-layered crystals at (a) 0° tilt and (b) 60° tilt. The circles in (a) indicate spots (16,15) corresponding to a resolution of 2.96 Å. In (b), the circle indicates spot (16,0) at a resolution of 3.7 Å. Crystal samples were embedded in 2% (w/v) glucose on molybdenum grids covered with a carbon film that was previously evaporated onto mica and floated on the grid. Electron diffraction patterns were recorded at low electron doses (<5 é/Å^2) on a Gatan 2K×2K CCD camera with a Philips CM200 FEG operated at 200 kV. The 2D crystals of OG-solubilized SoPIP2;1 were grown in the presence of *Escherichia coli* polar lipids at an LPR of 0.3 by dialysis for three days against a buffer containing 20 mM Tris–HCl (pH 8), 100 mM NaCl, 50 mM $MgCl_2$, 2 mM DTT, 0.03% (w/v) NaN_3.

high-resolution images of tilted specimens were collected and combined to obtain a 3D potential map (Table 1).

One tetramer and the complete unit cell are shown in Figure 2(a) and (b), respectively, with one monomer highlighted. Seven rod-like structures are clearly recognizable. By comparison with the atomic model of human AQP1,[21] they can be assigned to transmembrane helices 1–6 and loops B and E that form half-helices that fold back into the membrane to meet in the middle.[7] The remarkable arrangement of the two crystal layers stacked together by a tongue-and-groove fit of the extracellular surface, which is facilitated by the shift of one layer with respect to the other, is shown in Figure 2(c). This is radically different from the head-to-head coaxial packing of tetramers in the AQP0 double-layered crystals.[9]

The overall architecture of the channel has substantial similarity to AQP1, but some differences can be distinguished. The most remarkable difference is the length and straightness of helix 1. When compared to AQP1, helix 1 of SoPIP2;1 appears to protrude farther out from the extracellular membrane surface and to point towards the 4-fold axis of the tetramer (Figure 3(a) and (b)). A corresponding alteration is found for helix 2, which is more tilted at its N terminus than helix 2 of AQP1. All other helices fit well to the corresponding helices in AQP1. The visible differences in helices 1 and 2 dictate a different position of the connecting loop A in SoPIP2;1. In AQP1, this loop lies on top of helices 1 and 2, parallel with the side of the monomer. In the map of SoPIP2;1, however, the positions of the helix 1 C terminus and the helix 2 N terminus suggest that loop A is oriented towards the 4-fold center of the tetramer (Figure 3(a)). The tongue-and-groove packing arrangement results from the extracellular end of helix 1, which protrudes into the opposite layer, filling the gap between adjacent tetramers (Figure 2(b) and (c)). This suggests that the stability of the double-layered crystals might be the result of a contact between loops A from one layer with helices 3 and/or 6 of the opposing layer. Loop A in SoPIP2;1 is also different from that in AQP0, where tetramers are coaxially stacked face-to-face by prominent interactions between Pro38 in loop A, and a Pro–Pro motif (Pro109 and Pro110) in loop C.[9] It transpires that the crystalline packing of the two layers is, in both cases, dictated to a significant extent by the configuration of loop A.

Beside SoPIP2;1 and AQP0 a third aquaporin, AQP2, has been reported to preferably form double-layered two-dimensional crystals ordered to superior resolution.[22] In the case of AQP0,[9] this crystal form reflects the native state in the eye lens core, where AQP0 forms membrane junctions. AQP2 forms crystal contacts between the cytosolically located termini of the proteins and thereby differs from AQP0 and SoPIP2;1. The crystal packing of AQP2 is most probably not representing an *in vivo* situation but rather a crystallization artifact.[22] Whether the interactions of the extracellular parts of opposing SoPIP2;1 tetramers represent an *in vivo* situation has not been assessed. However, there are *in vivo* situations where different membrane domains of the plasma membrane of a cell are likely to interact *via* protein–protein interactions. Stomata closure and opening is accompanied by a substantial volume change of the guard cells. The membrane surface increase upon stomata opening cannot be explained by stretching of the plasma membrane.[23–25] It has to be attributed either to vesicles fusing with the plasma membrane or to plasma membrane invaginations being unfolded. Very fast cell volume changes necessitating vesicle fusion or invaginations unfolding have been recorded for mesophyll protoplasts isolated from leaf tissue.[26–28] Immunogold electron microscopy has demonstrated PIP subfamily members to be associated with plasmalemmasomes, invaginations of the plasma membrane protruding into the cytosol towards the central vacuole of *Arabidopsis* mesophyll cells.[29] It is possible that major integral proteins of the plasma membrane, such as

Table 1. Crystallographic data

Plane group symmetry	$p4$
Unit cell parameters	
$a=b$ (Å)	65
c (Å)	200 (assumed)
$\alpha=\beta=\gamma$ (deg.)	90
No. processed images	156 (0°, 21; 10°, 28; 15°, 16; 20°, 23; 30°, 37; 45°, 28; 60°, 3)
No. merged phases	24,479
Resolution limit for merging	5.0 Å (in the membrane plane; x,y-direction) 7.14 Å (perpendicular to the membrane plane; z-direction)
Phase residual (IQ-weighted) (deg.)	43.0 (overall) 34.4 (100–9.7 Å) 39.6 (9.7–6.9 Å) 50.6 (6.9–5.6 Å) 60.8 (5.6–5 Å)
Completeness[a] (%)	36.5 (resolution volume: 5 Å in x,y, 7.14 Å in z) 57 (resolution volume 6 Å in x,y, 7.14 Å in z) 65 (resolution volume 6 Å in x,y, 10 Å in z)

[a] Only reflections within the resolution volume having a figure of merit over 0.5 were included; the missing cone is comprised in this volume.

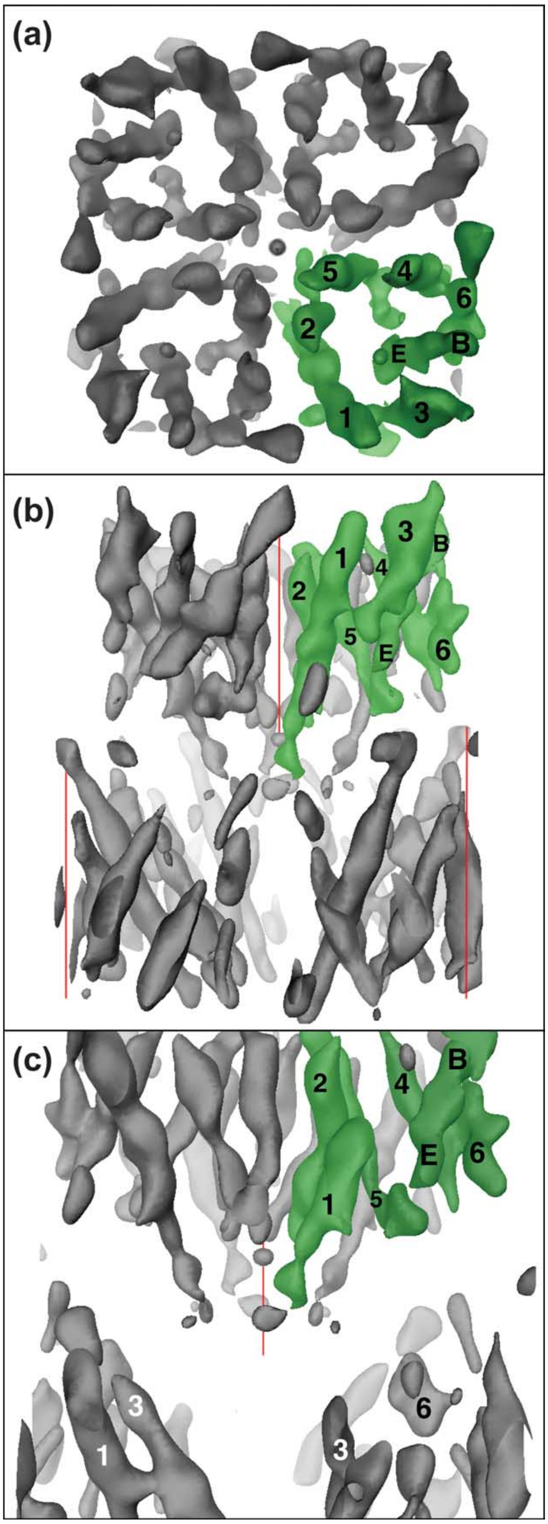


Figure 2. A 3D map of SoPIP2;1, calculated from 156 electron micrographs (see Table 1). (a) Cytosolic view of one tetramer. (b) Side-view of the unit cell comprising one tetramer and four monomers of the opposite layer. One monomer is highlighted in green. Helices 1–6 as well as loops B and E that fold back into the membrane and form a seventh transmembrane domain can be assigned as indicated. The 4-fold axes are drawn in red. (c) View into the center of the unit cell, where the extracellular ends of helices 1 from one tetramer protrude into the cleft between adjacent tetramers of the opposite layer. For clarity, parts of the map are cut away and the respective helices indicated. The map was calculated using the MRC software package.[32] Images were corrected for lattice distortions taking the Fourier-filtered images themselves as references. Measured phases and amplitudes were corrected for the tilted contrast transfer function and merged imposing *p*4 symmetry. Phase origins were refined and lattice lines for amplitudes and phases were fitted to create a 3D data set.

aquaporins, participate in interactions stabilizing membrane invaginations.

At the C terminus of loop A, PIPs exhibit a highly conserved Cys, which is not present in other aquaporin-like proteins in plants (Figure 3(c); U.J., unpublished results). Using the multiple sequence alignment reported by Heymann & Engel,[30] the conserved Cys of SoPIP2;1 was found to correspond to Thr44 in human AQP1 (Figure 3(c)). As displayed in Figure 3(a) and (b), the N-terminal part of the AQP1 helix 2 should be tilted towards the 4-fold axis to fit to the potential map of SoPIP2;1 in this region. The corresponding four Cys in SoPIP2;1 would then be remarkably close to each other, and close to the 4-fold axis of the tetramer. The high level of conservation suggests a special function for this Cys. According to their nearness, which is due to the particular configuration of loop A, these four Cys may stabilize the SoPIP2;1 tetramer by fostering hydrogen bonds or complexing a metal ion.

The particular arrangement of the four Cys was confirmed quantitatively by fitting helical segments to the 3D potential map of SoPIP2;1 using the program ROTTRANS.[31] In addition, the helical backbones of the atomic models of AQP1 and AQP0 in both open and closed conformation were fitted to the potential map of SoPIP2;1 and yielded cross-correlation scores of 0.37 (AQP1),[21] 0.366 (AQP0 open),[10] and 0.367 (AQP0 closed).[9] The question of whether the water channel of SoPIP2;1 is in an open or closed state cannot be answered at the resolution available. Differences are likely to be related to the conformation of single amino acid side-chains, such as the Tyr149 in AQP0.[9]

The high level of water permeability showed in activity measurements of proteoliposomes[19] suggests the water channels of SoPIP2;1 to be in an open conformation. However, the phosphorylation state of crystallized SoPIP2;1 is not known, and the mechanism of phosphorylation-mediated gating of the channel remains to be elucidated. The 2D crystals and the 3D map of the functional protein in a lipid bilayer provide a solid framework for this goal. As a result of the orientation of the tetramers within the double-layered crystals, the cytosolic C terminus as well as the B-loop are accessible for phosphorylation experiments, allowing the channel to be resolved in the open and closed states.

References

1. Johansson, I., Karlsson, M., Johanson, U., Larsson, C. & Kjellbom, P. (2000). The role of aquaporins in cellular and whole plant water balance. *Biochim. Biophys. Acta*, **1465**, 324–342.
2. Johanson, U., Karlsson, M., Johansson, I., Gustavsson, S., Sjövall, S., Fraysse, L. *et al*. (2001). The complete set of genes encoding major intrinsic proteins in *Arabidopsis* provides a framework for a new nomenclature for major intrinsic proteins in plants. *Plant Physiol.* **126**, 1358–1369.
3. Chaumont, F., Barrieu, F., Wojcik, E., Chrispeels, M. & Jung, R. (2000). Plasma membrane intrinsic proteins from maize cluster in two sequence subgroups with differential aquaporin activity. *Plant Physiol.* **122**, 1025–1034.
4. Johansson, I., Larsson, C., Ek, B. & Kjellbom, P. (1996). The major integral proteins of spinach leaf plasma membranes are putative aquaporins and are phosphorylated in response to Ca^{2+} and apoplastic water potential. *Plant Cell*, **8**, 1181–1191.
5. Johansson, I., Karlsson, M., Shukla, V. K., Chrispeels, M. J., Larsson, C. & Kjellbom, P. (1998). Water transport activity of the plasma membrane aquaporin SoPIP2;1 is regulated by phosphorylation. *Plant Cell*, **10**, 451–459.
6. Kjellbom, P., Larsson, C., Johansson, I., Karlsson, M. & Johanson, U. (1999). Aquaporins and water homeostasis in plants. *Trends Plant Sci.* **8**, 308–314.
7. Murata, K., Mitsuoka, K., Hirai, T., Walz, T., Agre, P., Heyman, B. *et al*. (2000). Structural determinants of water permeation through aquaporin-1. *Nature*, **407**, 605–612.
8. Sui, H., Han, B. G., Lee, J. K., Walian, P. & Jap, B. K. (2001). Structural basis of water-specific transport through AQP1 water channel. *Nature*, **414**, 872–878.
9. Gonen, T., Sliz, P., Kistler, J., Cheng, Y. & Walz, T. (2004). Aquaporin-0 membrane junctions reveal the structure of a closed water pore. *Nature*, **429**, 193–197.
10. Harries, W. E., Akhavan, D., Miercke, L. J., Khademi, S. & Stroud, R. M. (2004). The channel architecture of aquaporin 0 at a 2.2-Å resolution. *Proc. Natl Acad. Sci. USA*, **101**, 14045–14050.

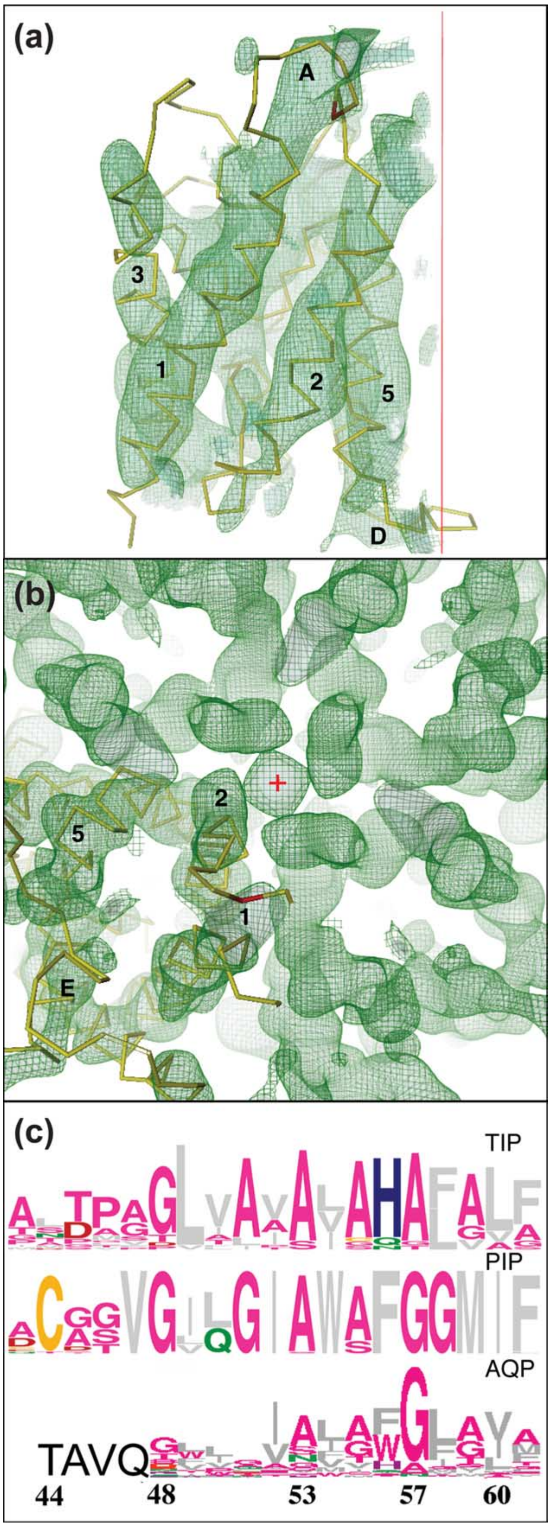


Figure 3. (a) Superposition of SoPIP2;1 potential map (green) with the C^{α} backbone of the atomic model of AQP1 (yellow); Thr44 is highlighted in red. The 4-fold axis is drawn in red in this side-view of the superimposed monomers, extracellular face up. (b) Extracellular view of a SoPIP2;1 tetramer with one AQP1 monomer fitted into a SoPIP2;1 monomer. The 4-fold center is indicated by a red cross. Superposition of the two structural datasets was performed using the program DINO (http://www.dino3D.org). (c) Alignment of the PIP and TIP consensus sequences of parts of loop A and helix 2 with the aquaporin consensus sequence of helix 2 (residues 48–61).[30] The PIP consensus sequence is based on SoPIP2;1 (AAA99274), all *Arabidopsis* PIPs,[2] maize PIPs,[33] and PIPs from *Picea abies* and *Physcomitrella patens* (CAB06080, CAB07783, AAS65964, translation of Physcobase contig 10071 http://moss.nibb.ac.jp/). The TIP consensus sequence is derived from *Arabidopsis*[2] and maize TIPs.[33] Sequence conservation is displayed by the sequence logos technique.[34,35] According to this alignment, the highly conserved Cys in PIPs corresponds to Thr44 in AQP1 (black letters).

11. Fu, D., Libson, A., Miercke, L. J. W., Weitzman, C., Nollert, P., Krucinski, J. & Stroud, R. M. (2000). Structure of a glycerol-conducting channel and the basis for its selectivity. *Science*, **290**, 481–486.
12. Savage, D. F., Egea, P. F., Robles-Colmenares, Y., O'Connell, J. D., III & Stroud, R. M. (2003). Architecture and selectivity in aquaporins: 2.5 Å X-ray structure of aquaporin Z. *PloS Bio.* **1**, 334–340.
13. De Groot, B. L. & Grubmüller, H. (2001). Water permeation across biological membranes: mechanism and dynamics of aquaporin-1 and GlpF. *Science*, **294**, 2352–2356.
14. Tajkhorshid, E., Nollert, P., Jensen, M. O., Miercke, L. J., O'Connell, J., Stroud, R. M. & Schulten, K. (2002). Control of the selectivity of the aquaporin water channel family by global orientational tuning. *Science*, **296**, 525–530.
15. De Groot, B. L., Frigato, T., Helms, V. & Grubmüller, H. (2003). The mechanism of proton exclusion in the aquaporin-1 water channel. *J. Mol. Biol.* **333**, 279–293.
16. Chakrabarti, N., Tajkhorshid, E., Roux, B. & Pomes, R. (2004). Molecular basis of proton blockage in aquaporins. *Structure (Camb.)*, **12**, 65–74.
17. Chakrabarti, N., Roux, B. & Pomes, R. (2004). Structural determinants of proton blockage in aquaporins. *J. Mol. Biol.* **343**, 493–510.
18. Daniels, M. J., Chrispeels, M. J. & Yeager, M. (1999). Projection structure of a plant vacuole membrane aquaporin by electron cryo-crystallography. *J. Mol. Biol.* **294**, 1337–1349.
19. Karlsson, M., Fotiadis, D., Sjövall, S., Johansson, I., Hedfalk, K., Engel, A. & Kjellbom, P. (2003). Reconstitution of water channel function of an aquaporin overexpressed and purified from *Pichia pastoris*. *FEBS Letters*, **537**, 68–72.
20. Fotiadis, D., Hasler, L., Müller, D. J., Stahlberg, H., Kistler, J. & Engel, A. (2000). Surface tongue-and-groove contours on lens MIP facilitate cell–cell adherence. *J. Mol. Biol.* **300**, 779–789.
21. De Groot, B. L., Engel, A. & Grubmüller, H. (2001). A refined structure of human aquaporin-1. *FEBS Letters*, **504**, 206–211.
22. Schenk, A. D., Werten, P. J. L., Scheuring, S., de Groot, B. L., Müller, S. A., Stahlberg, H. *et al.* (2005). The 4.5 Å structure of human AQP2. *J. Mol. Biol.* In the press.
23. Blatt, M. R. (2000). Cellular volume control in stomatal movements in plants. *Annu. Rev. Cell Dev. Biol.* **16**, 221–241.
24. Geelen, D., Leyman, B., Batoko, H., Di Sansebastiano, G. P., Moore, I. & Blatt, M. R. (2002). The abscisic acid-related SNARE homolog NtSyr1 contributes to secretion and growth: evidence from competition with its cytosolic domain. *Plant Cell*, **14**, 387–406.
25. Pratelli, R., Sutter, J. U. & Blatt, M. R. (2004). A new catch in the SNARE. *Trends Plant Sci.* **9**, 187–195.
26. Ramahaleo, T., Morillon, R., Alexandre, J. & Lassalles, J. P. (1999). Osmotic water permeability of isolated protoplasts. Modification during development. *Plant Physiol.* **119**, 885–896.
27. Morillon, R., Lienard, D., Chrispeels, M. J. & Lassalles, J. P. (2001). Rapid movements of plants organs require solute–water cotransporters or contractile proteins. *Plant Physiol.* **127**, 720–723.
28. Moshelion, M., Moran, M. & Chaumont, F. (2004). Dynamic changes in the osmotic water permeability of protoplast plasma membrane. *Plant Physiol.* **135**, 2301–2317.
29. Robinson, D. G., Sieber, H., Kammerloher, W. & Schaffner, A. R. (1996). PIP1 aquaporins are concentrated in plasmalemmasomes of *Arabidopsis thaliana* mesophyll. *Plant Physiol.* **111**, 645–649.
30. Heymann, J. B. & Engel, A. (2000). Structural clues in the sequence of the aquaporins. *J. Mol. Biol.* **295**, 1039–1053.
31. de Groot, B. L., Heymann, J. B., Engel, A., Mitsuoka, K., Fujiyoshi, Y. & Grubmüller, H. (2000). The fold of human aquaporin 1. *J. Mol. Biol.* **300**, 987–994.
32. Crowther, R. A., Henderson, R. & Smith, J. M. (1996). MRC image processing programs. *J. Struct. Biol.* **116**, 9–16.
33. Chaumont, F., Barrieu, F., Wojcik, E., Chrispeels, M. J. & Jung, R. (2001). Aquaporins constitute a large and highly divergent protein family in maize. *Plant Physiol.* **125**, 1206–1215.
34. Crooks, G. E., Hon, G., Chandonia, J. M. & Brenner, S. E. (2004). WebLogo: a sequence logo generator. *Genome Res.* **14**, 1188–1190.
35. Schneider, T. D. & Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucl. Acids Res.* **18**, 6097–6100.

Edited by W. Baumeister

(Received 18 February 2005; received in revised form 25 April 2005; accepted 2 May 2005)

2.4 6 Å Projection Map of Oligogalacturonate Porin KdgM of *Erwinia chrysanthemi*

Contribution

The unpublished work presented here is part of the PhD thesis of Giani Signorell. My contribution to this project is solely focused on the image processing. The purification, crystallization and microscopy was carried out by Giani Signorell in partial collaboration with Hervé Remigy and Mohamed Chami, and is included here only to give a short overview of the project. My contribution to the image processing of this project was the evaluation and processing of the recorded projection images and the merging of the projection data, which was collaboratively carried out with Giani Signorell. In addition, I performed the secondary structure prediction and the orientational correlation of the single pore.

2.4.1 Introduction

Located in the outer membrane of the Gram-negative bacteria *Erwinia chrysanthemi,* KdgM is responsible for transport of pectin degradation products across the outer membrane. Secondary structure prediction and site directed mutagenesis lead to the assumption that KdgM is a monomeric β-barrel composed of 14 antiparallel β-strands.

Crystallization: KdgM was crystallized by dialysis into crystalline tubes. The purified protein was provided by B. Condemine[1]. Briefly, the protein was solubilized in 1% LDAO and mixed with DMPC (solubilized at 5 $\frac{mg}{ml}$ in LDAO) to a final LPR of 0.1 to 0.15. The crystallization trials were conducted in dialysis buttons of 60 μl, covered with a 10 kDa-cutoff dialysis membrane and dialyzed by submerging in 3-l flasks containing a 20 mM Tris-HCl buffer at pH 9 with 200 mM NaCl and 0.03% NaN_3. The samples were dialyzed for 3 weeks at room temperature.

The dialysis yielded crystalline tubes with a diameter of up to 1 μm.

Image Acquisition: Negatively stained samples were prepared using uranyl acetate as stain. Images were recorded on Kodak SO163 film with a Hitachi H7000 electron microscope at a acceleration voltage of 100 kV.

---

[1] Unité Microbiologie et Génétique, UMR-CNRS-INSA-UCB 5122, Villeurbanne,1 Groupe Canaux Ioniques, France

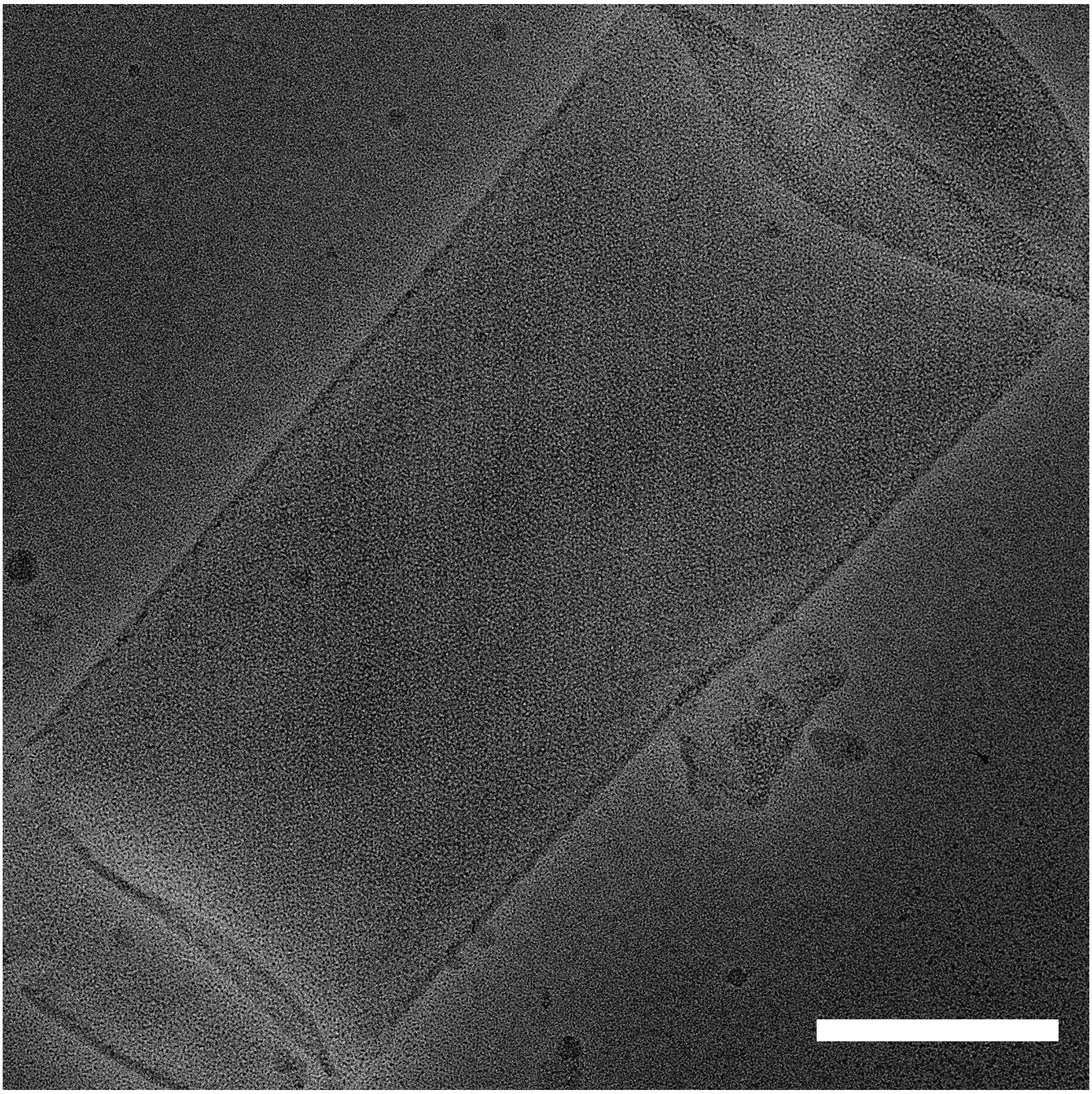


Figure 2.5: Tubular crystal of KdgM. The scale bar represents 250 nm.

For data recording KdgM crystals were embedded in trehalose, following the standard protocol for embedding (4.1.1), using a 2% Trehalose solution and omitting the second layer. Images were recorded at liquid nitrogen temperature on a Philips CM200FEG at an acceleration voltage of 200 kV and a nominal magnification of 50'000 ×. The negatives recorded on Kodak SO163 film were scanned at a resolution of 2000 $\frac{lines}{cm}$, resulting in a resolution of 1 $\frac{\text{Å}}{px}$ on the sample level.

2.4.2 Image Processing

Untilted images of KdgM tubes were processed using the MRC software package. The epitaxial twinned lattice could easily be separated in Fourier space, and the two individual lattices of the upper and lower half of the flattened tube were processed independently. The lattices were indexed manually, and the optimized image processing scheme was used (see Fig. 3.1). The unit cell vectors were found to be a=40 Å, b=115 Å and γ=90°, leading to a unit cell consisting of 4 monomers.

Exploring possible crystallographic packing arrangements by ALLSPACE of the MRC suite, the best symmetry turned out to be $p22_12_1$ (see Table 2.1), which means that pairs of two monomers are packed in a upside down manner. The monomer shows the form of on elliptical pore. The ratio between major and minor axis was determined to be 1.2 and the rotation of the major axis to the x axis was found to be 39° and -39° respectively.

2.4.3 Subunit Composition

To obtain information of the subunit composition I carried out a secondary structure prediction, comparing the result obtained by different prediction algorithm, using the tools available at `npsa-pbil.ibcp.fr`(see Fig. 2.8). The algorithm used were PHD by Rost and Sander[39] based on a two-layered feed-forward neural network which was trained using 130 independent protein chains, the HNN and MRLC algorithms by Guermeur et al.[16] which works by combining result achieved by GOR IV[11] and SOPMA[12] using multivariate linear regression techniques, and the DSC algorithm by King and Sternberg[23] which works by decomposing secondary structure prediction into following basic concepts: residue conformational propensities, sequence edge effects, moments of hydrophobicity, position of insertions and deletions in aligned homologous sequence, moments of conservation, auto-correlation, residue ratios and secondary structure feedback effects. This choice of algorithms was made to include an as broad range as possible of different prediction methods.

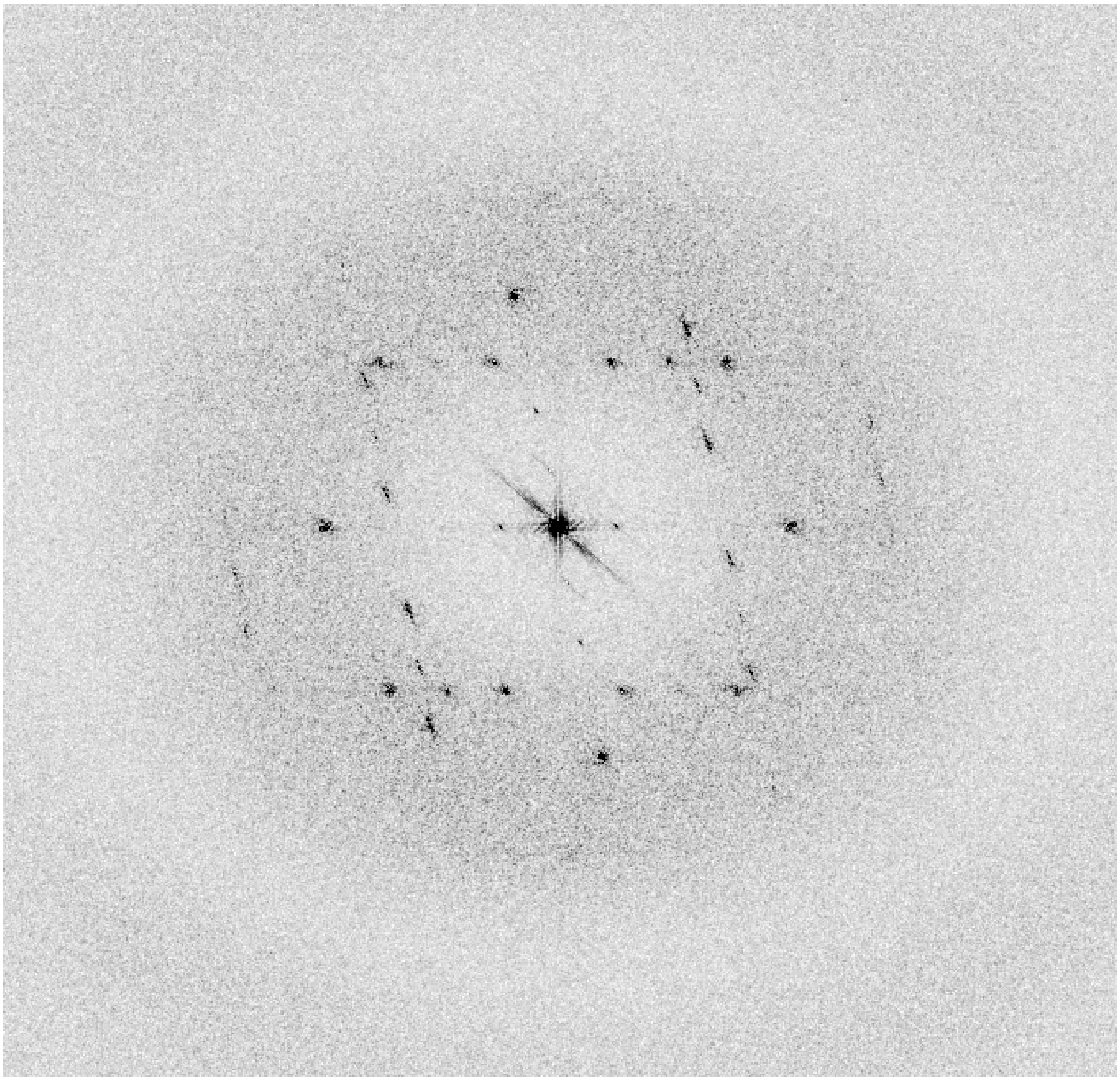


Figure 2.6: Fourier transform of 2.5 showing the two epitaxial twinned lattices

Symmetry	Phase Residual	Number of phases compared
p1	26.0	132
p2	37.2$^{(1)}$	66
p12_b	74.1	51
p12_a	75.4	44
p121_b	22.2$^{(1)}$	51
p121_a	19.5$^{(1)}$	44
c12_b	74.1	51
c12_a	75.4	44
p222	76.0	161
p2221b	62.2	161
p2221a	55.9	161
p22121	27.7$^{(1)}$	161
c222	76.0	161
p4	41.2$^{(3)}$	94
p422	76.2	215
p4212	34.3$^{(2)}$	215
p3	47.8	28
p312	66.9	89
p321	56.4	91
p6	50.3	122
p622	61.5	246

Table 2.1: Phase residual table for a KdgM projection for the different symmetries as determined by ALLSPACE. (1) acceptable (2) should be considered (3) possibility

Plane group symmetry	$p22_12_1$
Unit cell	a = 40 Å; b = 115 Å; c = 140 Å (assumed)
	$\gamma = 90°$
Number of processed images	10
Number of merged phases	900
Resolution limit for merging	6 Å
Phase residual (IQ-weighted)*	31.3° (Overall)
	21.9° (100 - 9.7 Å)
	35.7° (9.7 - 7.0 Å)
	61.2° (7.0 - 6.0 Å)
* = determined by AVRGAMPHS	

Table 2.2: Crystallographic data of KdgM

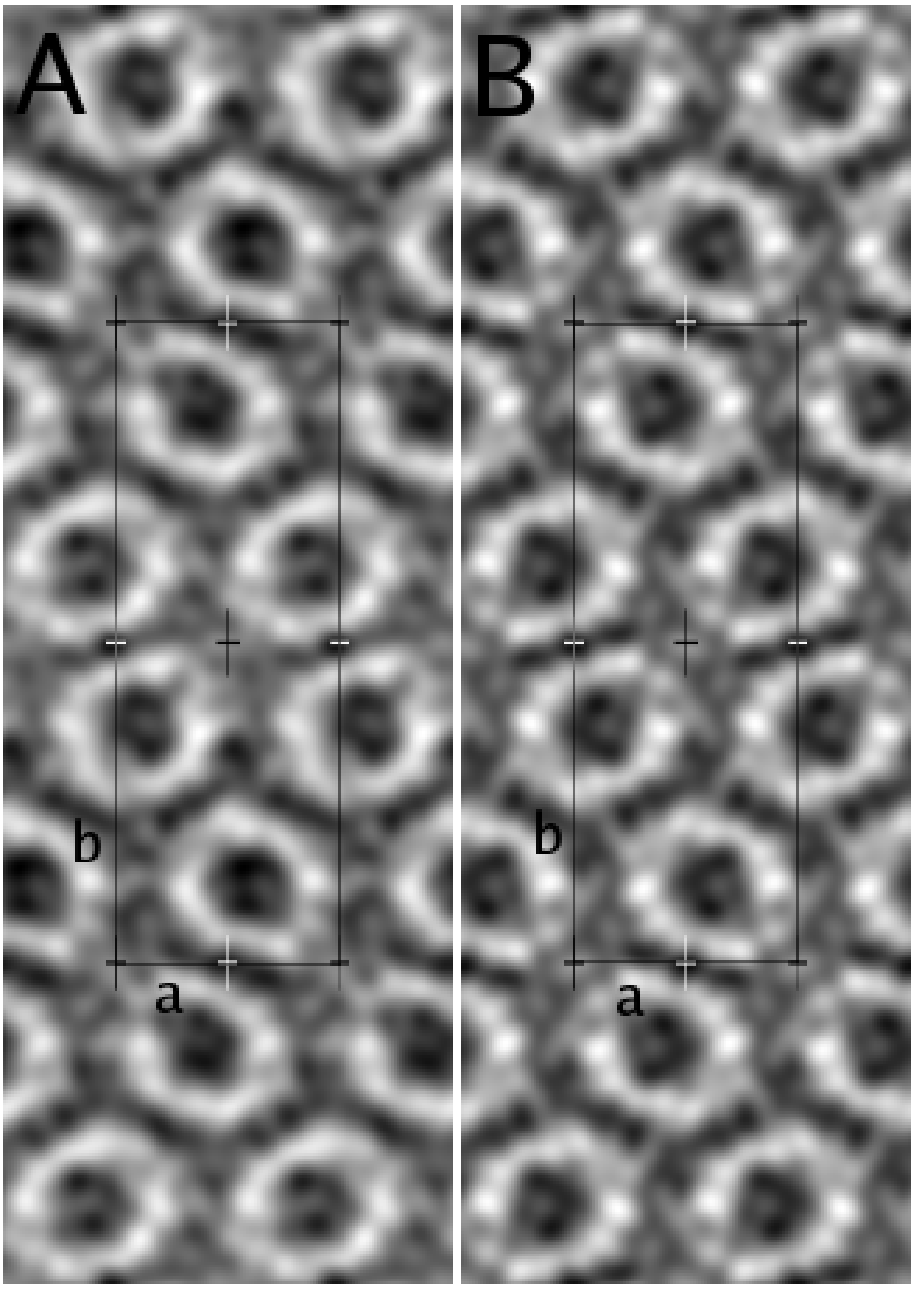


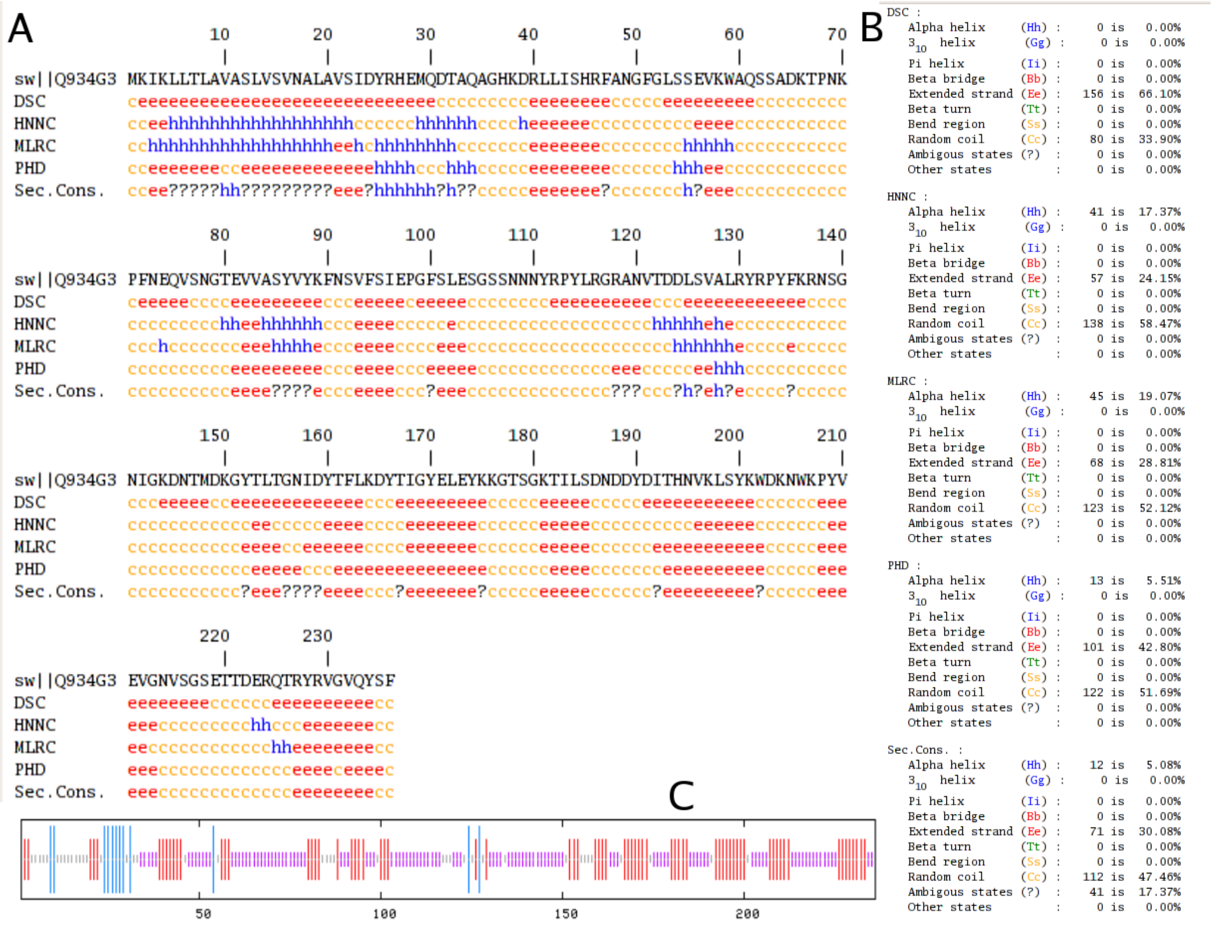
Figure 2.7: Averaged unit cell of KdgM. (A) unsymmetrized average (B) average after imposing $p22_12_1$symmetry. a = 40 Å; b = 115 Å; γ=90°

The secondary structure predictions suggests that KdgM consist of 14 β-strands. Site directed mutagenesis performed by Pellinen et al. 2003[33] comes to the same conclusion.

To identify the number of subunits composing the channel, which can clearly be seen in the averaged projection structure (Fig. 2.7), an oriental correlation of the ring part was performed. One of the symmetry related porins was cut out and the large and small axis of the ellipse were determined manually. After correcting for the axis difference the image was transformed in polar coordinates and the part responsible for building the ring was masked. The masked region was then averaged along the radius axis and the result was Fourier transformed. From the Fourier transform (Fig. 2.9) the periodicity, can be estimated. There is a strong signal for 3,9, and 18 fold periodicity but there are also good signals for a 13 fold and16 fold periodicity. The signal strength of different periodicities varied if the mask for the ring was changed, but a clear signal for a 14 fold periodicity, as would be expected from the secondary structure alignment, could never be observed. To explain this astonishing result one probably has to wait until tilted data of the KdgM crystals is available and a 3D reconstruction can be performed.

2.4.4 Outlook

A first set of tilted images at low tilt angle was recorded. The image processing of these images is in progress. The 3D dataset, which will be generated using these images, hopefully reveals some information on the arrangement of the β-sheets within the pore forming ring.



```
                 10        20        30        40        50        60        70
                  |         |         |         |         |         |         |
sw||Q934G3 MKIKLLTLAVASLVSVNALAVSIDYRHEMQDTAQAGHKDRLLISHRFANGFGLSSEVKWAQSSADKTPNK
DSC        ceeeeeeeeeeeeeeeeeeeeeeeeeeeeecccccccccceeeeeeeecccccceeeeeeeeeccccccccc
HNNC       cceehhhhhhhhhhhhhhhhhhcccccchhhhhhcccc heeeeeecccccccccceeeeccccccccccc
MLRC       cchhhhhhhhhhhhhhhhhheehchhhhhhhhccccccceeeeeeeccccccccchhhhhccccccccccc
PHD        cceeeeeeecceeeeeeeeeeeeehhhhccchhhccccceeeeeeeecccccchhheecccccccccccc
Sec.Cons.  ccee?????hh?????????eee?hhhhhh?h??ccccceeeeeee?ccccccch?eeeccccccccccc

                 80        90       100       110       120       130       140
                  |         |         |         |         |         |         |
sw||Q934G3 PFNEQVSNGTEVVASYVYKFNSVFSIEPGFSLESGSSNNNYRPYLRGRANVTDDLSVALRYRPYFKRNSG
DSC        ceeeeecccceeeeeeeeeccceeeeeceeeeecccccccceeeeeeeeeeccceeeeeeeeeeeecccc
HNNC       ccccccccchheehhhhhhccceeeecccccecccccccccccccccccccchhhhhehecccccccccccc
MLRC       ccchccccccceeehhhheccceeeecccceeeccccccccccccccccccccchhhhhheccccecccccc
PHD        cccccccccceeeeeeeeeccceeeecccceeeeeccccccccccccceeeccccceehhhcccccccccc
Sec.Cons.  cccccccccceeee????eccceeeeccc?eeecccccccccccccc???ccc?h?eh?ecccc?ccccc

                150       160       170       180       190       200       210
                  |         |         |         |         |         |         |
sw||Q934G3 NIGKDNTMDKGYTLTGNIDYTFLKDYTIGYELEYKKGTSGKTILSDNDDYDITHNVKLSYKWDKNWKPYV
DSC        ccceeeeecceeeeeeeeeeeeeeccceeeeeeeeecccccceeeeecccccceeeeeeeeeeecccccceee
HNNC       cccccccccccceeccccceeeecccceeeeeeecccccceeeeeccccccccccceeeeeeccccccceee
MLRC       ccccccccccceeeecceeeeeecccceeeeeeecccccceeeeecccccceeeeeeeeeeeccccceee
PHD        cccccccccccceeeeeccceeeeeeeeeeeeeeecccccceeeeccccccceeeeeeeeeecccccceee
Sec.Cons.  ccccccccccc?eee????eeeeccc?eeeeeee?ccccceeeeecccccc?eeeeeeeee?ccccceee

                220       230
                  |         |
sw||Q934G3 EVGNVSGSETTDERQTRYRVGVQYSF
DSC        eeeeeeeecccccceeeeeeeeeecc
HNNC       eeeccccccccchhccceeeeeeecc
MLRC       eecccccccccccchheeeeeeeecc
PHD        eeecccccccccccccceeeeceeeec
Sec.Cons.  eeeccccccccccccceeeeeeeecc
```

```
DSC :
   Alpha helix     (Hh) :     0 is   0.00%
   3_10 helix      (Gg) :     0 is   0.00%
   Pi helix        (Ii) :     0 is   0.00%
   Beta bridge     (Bb) :     0 is   0.00%
   Extended strand (Ee) :   156 is  66.10%
   Beta turn       (Tt) :     0 is   0.00%
   Bend region     (Ss) :     0 is   0.00%
   Random coil     (Cc) :    80 is  33.90%
   Ambigous states (?)  :     0 is   0.00%
   Other states         :     0 is   0.00%

HNNC :
   Alpha helix     (Hh) :    41 is  17.37%
   3_10 helix      (Gg) :     0 is   0.00%
   Pi helix        (Ii) :     0 is   0.00%
   Beta bridge     (Bb) :     0 is   0.00%
   Extended strand (Ee) :    57 is  24.15%
   Beta turn       (Tt) :     0 is   0.00%
   Bend region     (Ss) :     0 is   0.00%
   Random coil     (Cc) :   138 is  58.47%
   Ambigous states (?)  :     0 is   0.00%
   Other states         :     0 is   0.00%

MLRC :
   Alpha helix     (Hh) :    45 is  19.07%
   3_10 helix      (Gg) :     0 is   0.00%
   Pi helix        (Ii) :     0 is   0.00%
   Beta bridge     (Bb) :     0 is   0.00%
   Extended strand (Ee) :    68 is  28.81%
   Beta turn       (Tt) :     0 is   0.00%
   Bend region     (Ss) :     0 is   0.00%
   Random coil     (Cc) :   123 is  52.12%
   Ambigous states (?)  :     0 is   0.00%
   Other states         :     0 is   0.00%

PHD :
   Alpha helix     (Hh) :    13 is   5.51%
   3_10 helix      (Gg) :     0 is   0.00%
   Pi helix        (Ii) :     0 is   0.00%
   Beta bridge     (Bb) :     0 is   0.00%
   Extended strand (Ee) :   101 is  42.80%
   Beta turn       (Tt) :     0 is   0.00%
   Bend region     (Ss) :     0 is   0.00%
   Random coil     (Cc) :   122 is  51.69%
   Ambigous states (?)  :     0 is   0.00%
   Other states         :     0 is   0.00%

Sec.Cons. :
   Alpha helix     (Hh) :    12 is   5.08%
   3_10 helix      (Gg) :     0 is   0.00%
   Pi helix        (Ii) :     0 is   0.00%
   Beta bridge     (Bb) :     0 is   0.00%
   Extended strand (Ee) :    71 is  30.08%
   Beta turn       (Tt) :     0 is   0.00%
   Bend region     (Ss) :     0 is   0.00%
   Random coil     (Cc) :   112 is  47.46%
   Ambigous states (?)  :    41 is  17.37%
   Other states         :     0 is   0.00%
```

Figure 2.8: Secondary structure prediction. The secondary structure prediction was carried out by comparing the PHD,HNN,MRLC and DSC algorithm. (A) Prediction result of the individual algorithms. (B) Secondary structure composition listed for every algorithm. (C) Most likely structure predicted by comparing the different algorithms.

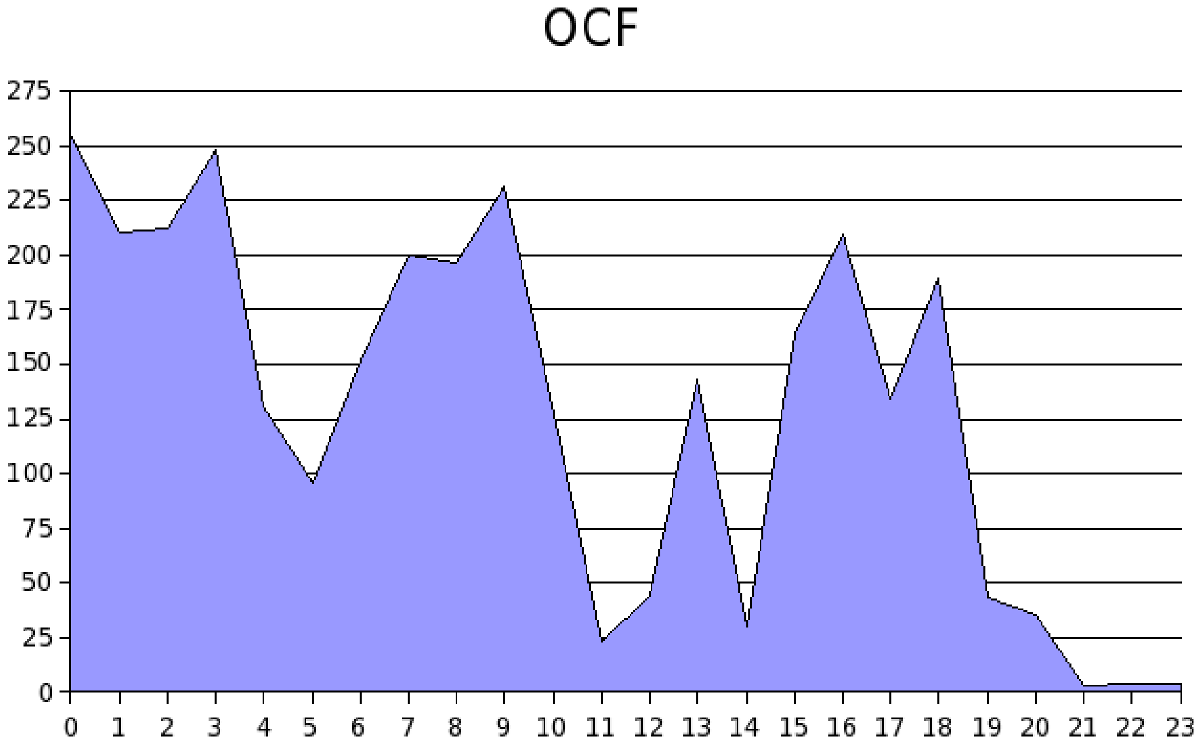


Figure 2.9: Rotational power spectrum

Chapter 3

New Developments in Image Processing

3.1 Introduction

3.1.1 Processing of Cryo-EM Images

The image processing scheme used for processing cryo images of AQP2, SoPIP2;1 and KdgM can be seen in Figure 3.1. It follows for large parts the original scheme used in MRC[18], but at several points the processing algorithms were modified as explained in detail in section 3.2, in order to improve the overall processing quality. Our modified scheme also includes two additional steps not present in the established scheme, namely an additional automated masking step and the cyclical refinement using a discretized 3D dataset (see 3.2.5 and 3.4.10 for details).

The established procedure used for processing 2D electron crystallographic images in MRC consists of following steps:

1. *Manual Masking:* The image is masked manually to include only the crystalline regions.
2. *Determination of the Crystal Lattice:* The lattice is manually determined using the visualization program XIMDISP. One drawback is the historic restriction to 8 bit displays. In XIMDISP, individual spots are clicked and their respective indices are entered manually. From the clicked spot, XIMDISP calculates the lattice, fitting the entered indices best.

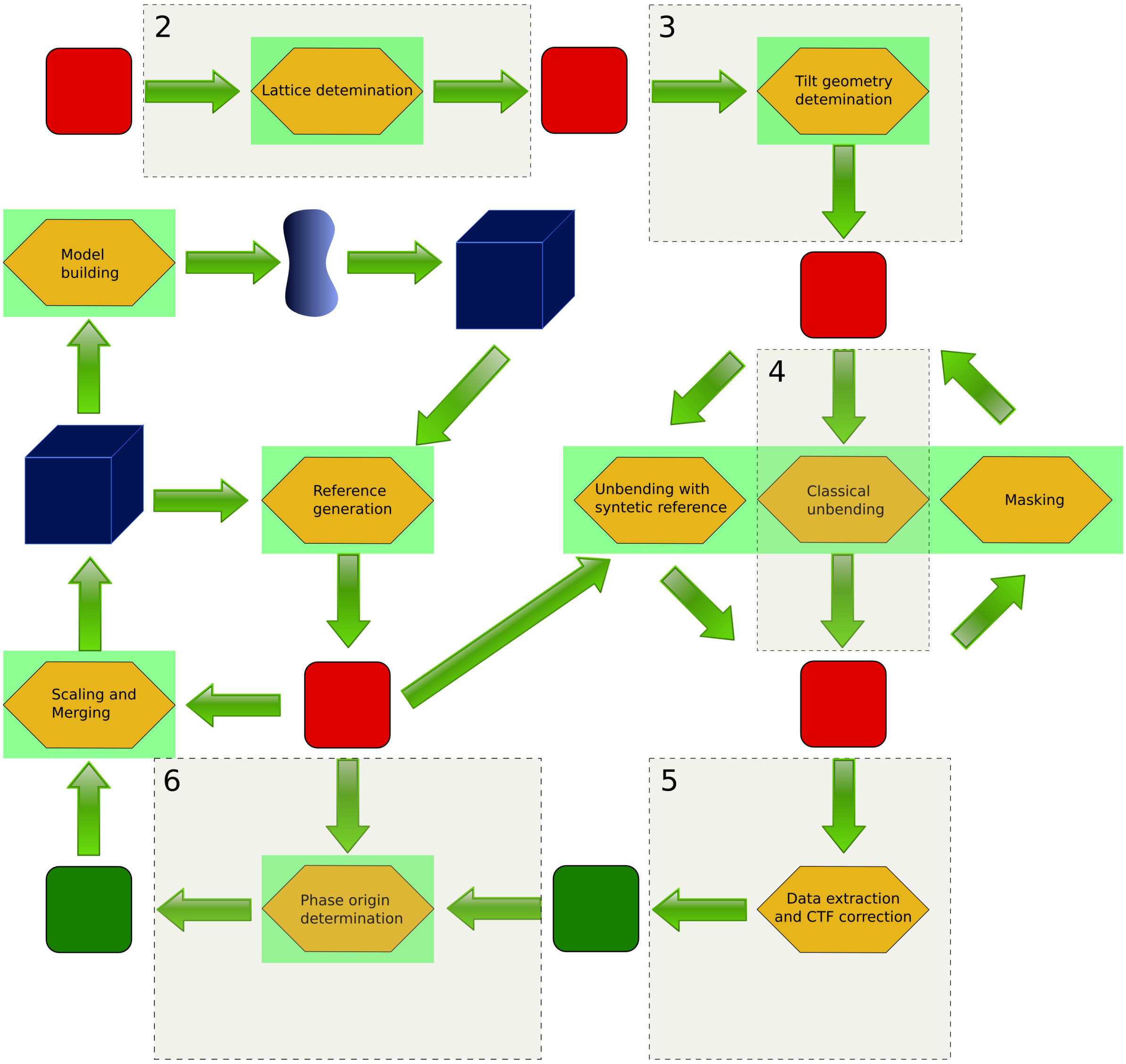

2
Lattice detemination
3
Tilt geometry detemination
Model building
4
Reference generation
Unbending with syntetic reference
Classical unbending
Masking
Scaling and Merging
6
5
Phase origin determination
Data extraction and CTF correction

Figure 3.1: Processing scheme for electron crystallographic images. Images are represented as red squares, program steps as yellow hexagons reflection data as green squares and three dimensional data in blue. The steps placed against a green background indicate modifications to the processing scheme that were implemented in the frame of this thesis to improve processing speed and/or quality.

3. *Tilt Geometry Determination:* The tilt geometry is calculated using Shaw's algorithm[40] by determining the distortions of the previously determined reciprocal lattice.

4. *Correction for Crystal Distortion:* The most straight forward way to correct for bends is to use the image itself as reference for searching displacements in individual unit cells. For this, the image is Fourier filtered by a tight mask (MASKTRAN) including only the very center of the spots. Additionally, a spotlist is used to restrict the mask to good spots which are determined manually. This reference is then cross correlated with a broadly masked image (MASKTRAN), which thus still contains the information of the lattice distortion.

   The cross correlation image is then searched by QUADSERCH for peaks corresponding to the individual unit cells. QUADSERCH starts from the center and tries to predict the expected position of a peak taking the known lattice and the already found distortions into account. In the close proximity of the predicted position a peak search for a cross correlation peak is then performed to find the actual position of the unit cell. From the difference of nominal position of the unit cell given by the lattice and the actually position found by the peak search, the displacement vector for this unit cell is calculated.

5. *Correction for the Contrast Transfer Function:* For a known transfer function of an untilted specimen the correction is usually performed using a deconvolution with a Wiener filter due to the noise in the image. For a tilted sample the the measured data cannot be explained as convolution of sample data and a transfer function of the microscope; as a consequence, different means of correction have to be found. As a first approximation, the tilt can be ignored for slightly tilted images – but for highly tilted images this clearly distorts the result. The MRC program TTBOX solves this problem by a heuristic approach that locally convolutes the data with the expected transfer function for a second time, yielding a diffraction peak of twice the height expected instead of a split peak originally produced by the tilted transfer function. A mathematical approach, solving the problem of correction of a tilted transfer function, has recently been tackled as seen in Philippsen et al. (submitted)[35], hopefully allowing better means of correction.

6. *Phase Origin Determination:* The phase origin is determined either manually or by searching the phase origin with the help of ORIGTILT, comparing the averaged image data to a reference density. The problems involved with this approach are discussed in 3.3.2.

3.1.2 Processing of Electron Diffraction Patterns

The basic processing of a single electron diffraction pattern in MRC consist of following steps, as described in 1990 by Henderson et al.[19] (see also Fig. 3.2):

1. *Lattice Determination:* The lattice determination is carried out by AUTOINDEX, performing a peak search and then averaging the diffraction pattern by subsequently shifting the image center to the peak positions found and adding the newly shifted image to the already averaged ones.
2. *Origin Determination:* The origin is determined by the program BACKAUTO which calculates the direction of the biggest background gradient in a number of subimages and then determines the origin from the crossing point of the gradient vectors.
3. *Tilt Geometry Determination:* The tilt geometry is determined from the lattice distortions using the program EMTILT.
4. *Amplitude Extraction:* The amplitudes of the diffraction patterns are extracted using the program PICKAUTO, which calculates the peak volumes by summing up the peak in a given area and subsequently correcting it for the background contribution by measuring the background in nearby areas.

3.1.3 3D Reconstruction

Scaling: Once the individual diffraction patterns and images are processed, a scale factor has to be applied to each dataset to account for the different measuring conditions, such as illumination, thickness of the ice and carbon film, or gain of the CCD camera. There are two different strategies to calculate the scaling factor.

Scaling on a Common Line: The finite extent of a crystal in z-direction can be approximated by a crystal of infinite extent multiplied by a rectangle function, which in Fourier space translates to a convolution by a sync function (the Fourier transform of the rectangle function). The oscillation of the continuous function along the lattice line for a finite crystal is therefore limited by the frequency of the sync function, which is in turn dependent on the width of the rectangle function. As a consequence, for a small z^* range, the amplitude and phase on a lattice line are expected to change only marginally, hence amplitudes and phases within this range, coming from two diffraction

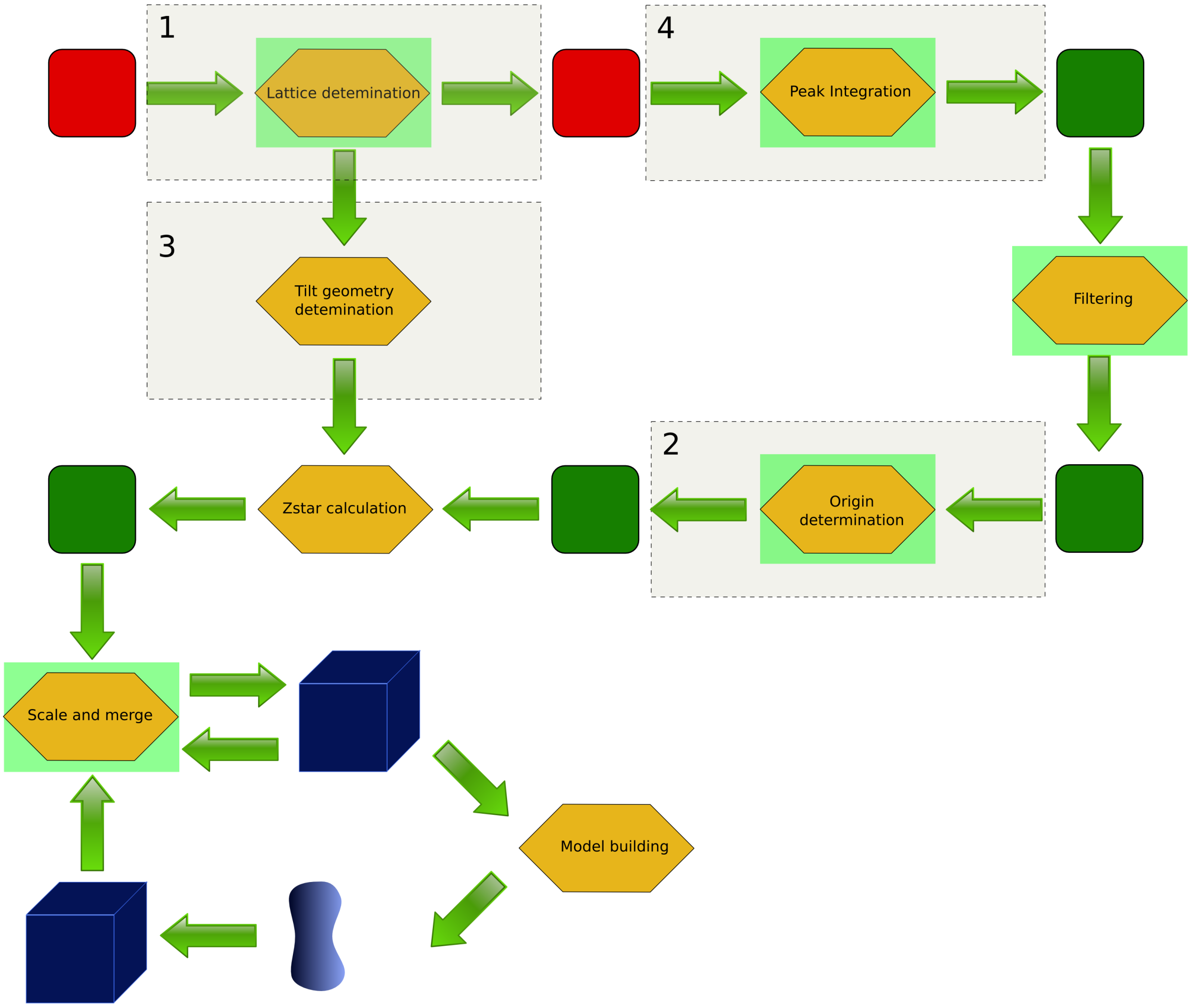


Figure 3.2: Processing scheme for electron diffraction patterns. Diffraction patterns are represented as red squares, program steps as yellow hexagons reflection data as green squares and three dimensional data in blue. The steps placed against a green background indicate modifications I applied to the processing scheme to improve processing speed or quality.

patterns or images, can be compared directly and the scaling factor can be calculated by a least squares fit.

Scaling Against a Reference Structure Instead of scaling the individual diffraction patterns or images against each other, all the data can also be scaled against a reference structure, if one is available. For this, a central section of the reference with the same tilt geometry as the to be scaled diffraction pattern or image has to be generated.

For images both strategies are implemented exploiting the different working modes of ORIGTILT to allow cross validation of the results obtained. Diffraction patterns are commonly scaled by MERGEDIFF.

Discretization

Prior to backtransforming the 3D Fourier dataset into real space, it has to be discretized along z^*. The MRC program LATLINE[2] fits a profile function along a lattice line and then samples the fit at a given interval, taking the phase restrictions of the given spacegroup into account.

3.1.4 Problems encountered

During the work at different 2D electron crystallographic projects (see chapter 2) it became evident that the current implementation of the image processing is cumbersome and that certain algorithms were missing or not suitable for the current projects.

Several problem areas were identified. On the high, user interaction level, passing of required meta-data to the relevant programs was solved suboptimally. For most programs the meta-data is entered on the standard input. For a program like ORIGTILT, which needs a lot of meta-data, this means entering 19 parameters concerning the merging and an additional 22 parameters for every file to be merged. Entering this meta-data manually is inefficient and in consequence most users write scripts feeding the parameters to the programs. For processing a second image, the script is normally copied to the new directory and the image specific parameters are changed. In case an erroneous parameter is found, the error has very likely already propagated to several image directories. A second problem, concerning the user interaction level, was identified to be the enormous amount of descriptive output generated during processing.

Although it is good to log all the image processing steps as precisely as possible, the user has a hard time to filter out the relevant information.

Another weakness in the MRC framework is the partially missing high level modularity. Some programs can carry out several different tasks, depending on the flags set, and this tends to baffle the novice user. As an example ORIGTILT can be used to merge a set of image data, but it can also be used to refine tilt parameters and phase origin for single images. Although combination of these two tasks can be understood by a developer, because both tasks include common sub steps, for a user not acquainted with the program code this is not at all obvious.

In the low level architecture, the same lack of modularity could also be observed at some places. Part of the subroutines used in the code are duplicated in several executables. For example, the subroutine ASYM can be found in the executables CTFSEARCH,ORIGTILT,TTREFINE,MAKETRAN and MERGEDIFF.

Difficulties were also encountered concerning the portability of the MRC package. Some of the programs are not available for all platforms, like the original version of SYNCFIT, which only runs on VMS. Part of the code, mostly the f77 subroutines from the Harwell library[1], is also not completely compatible with modern optimizing compiler like the Intel Fortran compiler on Linux.

Several programs were found to hide critical parameters in the source code, which forces the user to recompile and link the corresponding program if that parameter needs to be adjusted. Examples for this are THRESH and DIST in the program AUTOINDEX.

Various important functionalities used for processing the existing crystal data in our lab were missing in the MRC package. For example, some algorithms like detwinning of diffraction data or the Unix version of the SYNCFIT program were only implemented for certain symmetries. Some algorithms like automatic lattice determination for images, automated masking of crystalline regions or spotlist generation were completely missing.

Although it is clear that above mentioned points are in part consequences of the continual evolution of the MRC program package and cannot be completely avoided even by redesigning a new software from scratch, the amount of problematic code portions essentially makes it very challenging and cumbersome to add newly needed functionality in an efficient and transparent way.

As a consequence the image processing tools were partly modified and extended within MRC as much as possible, but in parallel a complete re-design and re-implementation effort was started with the iplt framework, in order to provide the necessary tools the solve the atomic structure of membrane proteins by 2D electron crystallography.

One goal was to reduce the amount of knowledge needed for using the tools and to present these tools in a way that an average user is not overwhelmed by the complexity of the interface. Another was to re-evaluate existing algorithms, tweaking them where appropriate, and also add novel ones.

The increasing data volume made it necessary to automate as many steps during processing as possible. Therefore, existing algorithms were made more robust and less sensitive to noise and measuring errors, and new algorithms were implemented to reduce the manual interaction needed during the original processing.

This development was in collaboration with the work done for automating image processing in MRC by H.Stahlberg (unpublished) and the novel software framework IPLT initiated by A. Philippsen[34] (see section 3.5). Section 3.4 describes the changes applying to both image and diffraction processing implemented by myself in the context of the processing framework as it is established in our group. Section 3.2 focuses on my developments to the part of the processing concerning images and section 3.3 summarizes the improvements done for diffraction processing.

3.2 New Developments in Processing Cryo Images

3.2.1 Spotlist

An epitaxial twinned crystal exhibits two lattices in the corresponding Fourier transform. After indexing one lattice it is necessary to ensure that spots where both lattices overlap are not included in the image processing. To this end, a small tool called SPOTSELECT, has been written which excludes overlapping spots within a user defined range from the spot list used for masking the Fourier transform prior to unbending. The tool needs (1) the two indexed lattices and (2) a pixel range giving the area within which overlapping spots should be excluded.

3.2.2 Image Preparation

Several algorithms in image processing, for example the cross correlation, scale with a complexity of $\Theta(N^2)$, which meant that on our available image processing hardware[1] the maximal reasonable image size in terms of processing speed was observed to be 8192×8192 pixel. Kodak SO-163 negatives have roughly the size of 9 cm ×6 cm which leads – depending on the exact size of the area scanned – to a digital image of around

---

[1] 1 SGI Origin 200 (4x270MHz) and 1 SGI Origin 2000 (4x250MHz)

16’000 ×12’000 pixel when scanned at a resolution of 2000 lines per cm. To automate the image processing, each picture is therefore automatically divided into 6 subpictures. This task is facilitated by a script running in the background, which takes every image within a special starting directory, converts it from negative transmission to OD, cuts it apart, creates the corresponding directory for the subimage, and creates a meta info file holding the basic data for the image (like size, name and nominal tilt angle). For the script to identify the image, a naming scheme was agreed upon for all images: PPPAANNNNSS.tif, where PPP is a three letter identifier for the processing project, AA is the nominal tilt angle, NNNN is the image number given by the microscope, and SS is the number of the subimage ranging from one to six. The script contains a load balancing mechanism, so that the CPU intensive tasks are only started when the load average of the used machine is below a certain threshold.

3.2.3 Tilt Geometry

To merge the datasets, generated from different images in Fourier space, it has to be known which central section each dataset represents, as defined by the tilt axis and tilt angle of the sample during recording the image. The tilt angle of the goniometer gives a crude estimate of the tilt angle in the image, and with some knowledge of the microscope lens system, the position of the tilt axis can be estimated. Unfortunately, these values are not precise enough for image processing, because the electron microscopy grids are not perfectly flat and the set tilt angle of the goniometer only roughly agrees with the measured tilt angle of the tip of the cryo holder, leading to a deviation of measured and nominal tilt geometry. It is therefore necessary to determine the tilt axis and tilt angle in a more precise way. One way to do so is to analyse the defocus change across an image.

To fit a contrast transfer function and, as a result, obtain the defocus value in different parts of an image proved to be more challenging than anticipated. Following reasons lead to a false fit or to no fit at all: (1) The carbon support film providing most of the contrast to the contrast transfer function was made as thin as possible to reduce the background signal in the measurement. (2) The second carbon layer placed on top of the grid to reduce charging in tilted samples naturally has a slightly different distance to the focal plane than the support film leading to two overlapping contrast transfer functions. (3) Within the subimage where a defocus measurement was taken the defocus was assumed to be constant in first approximation. For highly tilted images this approximation proved to be problematic but still necessary since CTFFIND was used for fitting.

To achieve a reasonable fit for the tilt plane parameters, albeit incorrect defocus fits (see Fig. 3.3B) a cyclic procedure was used to continually filter out the outliers in the defocus data, outlined as follows:

- The first step is to fit the CTF at different positions on the image: the image is subdivided into 7 × 7 subimages, and each of these subimages is then fitted individually by CTFFIND[15] (for an example see Fig. 3.3).
- Once every CTF has been determined, the defocus dz at all positions is known. Assuming a flat sample all the (x,y,dz) triplets should lie on a plane.
- The plane parameters are determined by the self written program TILTGEOM, which fits the optimal parameters for the defocus plane by a least squares algorithm.
- Points with a defocus deviating from the calculated defocus by more than a given threshold–3000 Å by default–are excluded from the next fit.
- The tilt plane parameters are now fitted repetitively, including all spots with a defocus deviation less than the given threshold. Every time the fit includes more than a given amount of points (16 by default) the threshold is reduced by multiplying it by a factor of 0.9. If the fit includes less than a given amount of points (12 by default) and is accordingly defined as unsuccessful, the threshold is increased by multiplying the threshold by a factor of 1.2.
- The repetitive fitting procedure stops after a given number (50 by default) of successful fits or after a total number (1000 by default) of total fits.
- The intersection line of the fitted plane and the x,y-plane is then calculated using the geometry functions of a self written vector geometry library giving the tilt axis. The tilt angle is determined by calculating the angle between the normal vectors of the determined plane and the the x,y-plane.

Improving the defocus determination could result in a more precise determination of the tilt geometry as it is heavily dependent on the measured defocus values. A CTF fitting routine implemented in IPLT, using the ACE algorithm[28], is currently in development. Incorporating a fit for a tilted contrast imaging function (TCIF)[35] within a subimage would yield the tilt geometry directly, as it is part of the formula.

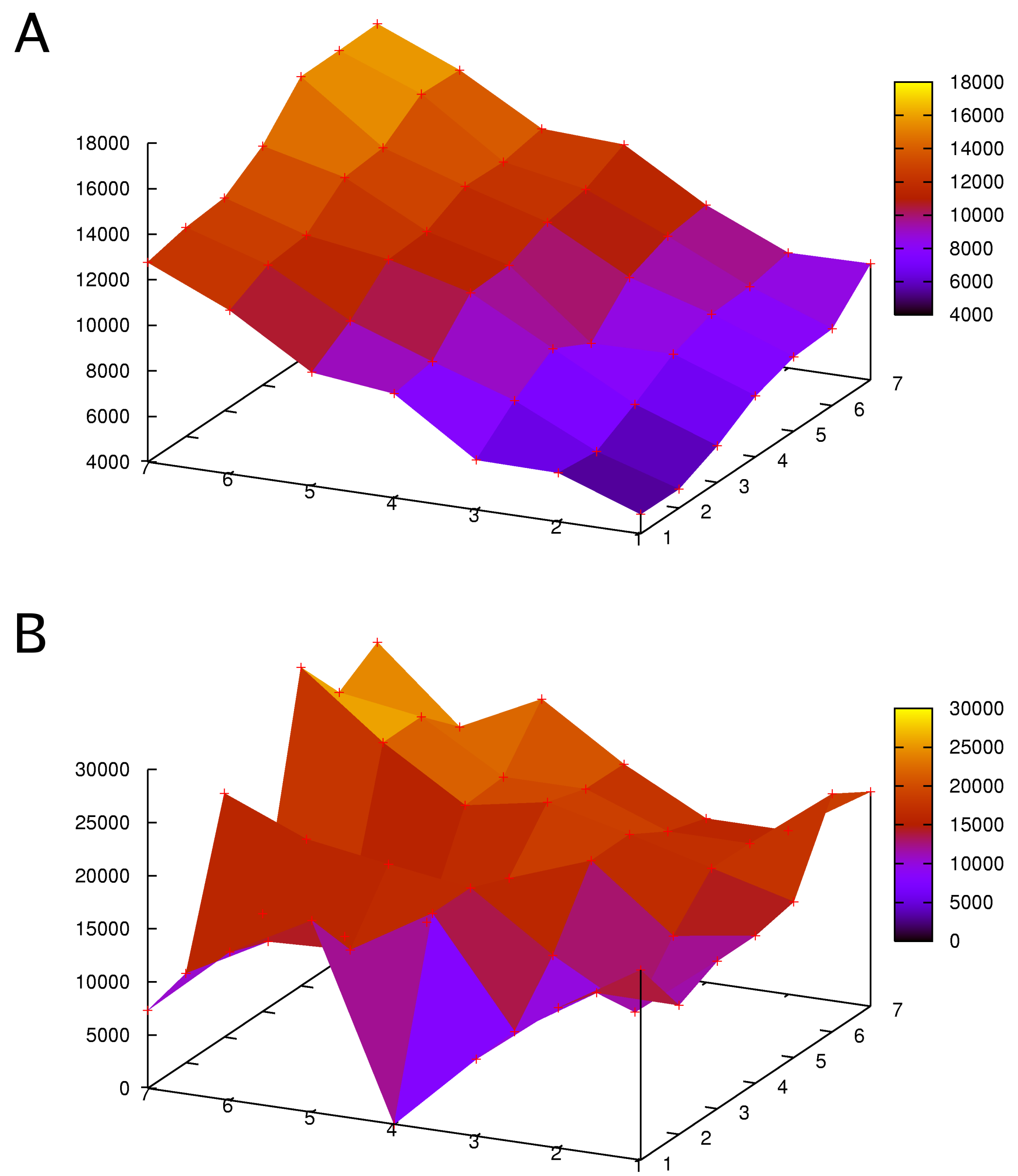


Figure 3.3: Fitted defocus values at 7 × 7 sub-image positions for two 60° tilted images. (A) Image with overall consistent defocus fits. (B) Image containing many outliers in the defocus fit.

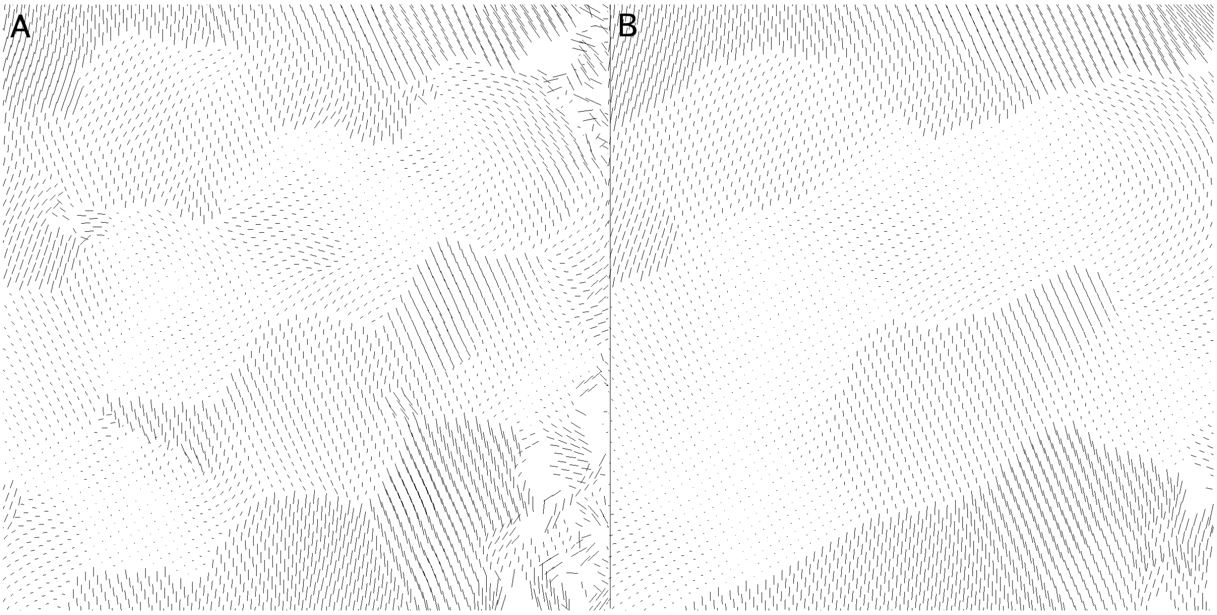


Figure 3.4: Displacement vectors during unbending with and without spot scan option. (A) Unbending with spot scan option deactivated (B) Unbending with spot scan option activated

3.2.4 Unbending

Two dimensional protein-lipid arrays are usually not perfectly ordered. In addition adsorption on the carbon film during sample preparation induces additional deformation in the crystal lattice. As a consequence correction for bends in the crystal lattice can improve the extracted high resolution data tremendously. The unbending is carried out by the two MRC programs QUADSERCH and CCUNBEND. Several modifications were integrated into QUADSERCH to improve unbending of Spot Scan images. In addition, alternatives to using a Fourier filtered reference for unbending were evaluated.

Unbending of Spot Scan Images with a Fourier Filtered Reference: The QUADSERCH program was modified to permit processing of spot scan images. If the lattice point to be searched is positioned at an edge of an illuminated spot scan spot, its exact position is predicted from the distortion found for the lattice points in the directly neighbouring spot scan spot, hence bridging overlapping and unilluminated areas between two spot scan spots, which give a low cross correlation signal. Figure 3.4 shows the difference in unbending between the unmodified and the modified QUADSERCH version.

Unbending using a Synthetic Reference: As demonstrated earlier by Kunji et al.[27] a reference image for unbending can be generated using a global reference structure. The approach used in MRC starts from a discretized reflection set. Having an arbitrary tilt axis and tilt angle, the measured dataset will normally not coincide with the reference dataset for all lattice lines. Consequently MAKETRAN interpolates in Fourier space, using a sum of sync functions to calculate amplitudes and phases at the needed positions.

A new approach was developed using a mass density as reference, implemented in IPLT. The reference image for unbending is calculated by explicit Fourier summation at exactly that central section used by the image, and therefore avoiding Fourier space interpolation. The central section is corrected by the according CTF before the dataset is backtransformed to real space, giving the projection of a single unit cell. The reference crystal is assembled and masked in real space. Although the real space operations for assembling and masking are more time consuming then their counterparts in Fourier space, they produce a superior result.

A comparison between the references for an untilted image generated by MAKETRAN and the new IPLT method can be seen in Fig 3.5. An artificial protein was taken as a test case. The protein was built in a fashion that it shows no symmetry and that the tilt- and axis-angle of projections can easily be seen by eye. The unit cell was set to $a = 90$, $b = 66$, $\gamma = 110$ and $c = 110$. Figures 3.5A and B clearly show that the sync interpolation in MAKETRAN produces unwanted artifacts in an image, whereas the Fourier summation implemented in IPLT (3.4.12) produces a much cleaner result. Preliminary results suggests an improvement in unbending quality using the new algorithm for reference generation, but more extensive tests have to be carried out to quantise the achieved improvements.

A problem of all references generated for unbending is the low cross correlation signal generated at the edge regions of tilted images, which is caused by the difference between the edge and center defocus. To circumvent this problem, a divide and conquer strategy could be implemented in the future, dividing an image into stripes with constant defocus and processing the stripes independently. Implementation of this strategy is still pending.

3.2.5 Masking

To improve the signal to noise ratio, the image is masked after the first unbending cycle to include only sufficiently crystalline regions. The correlation factor between

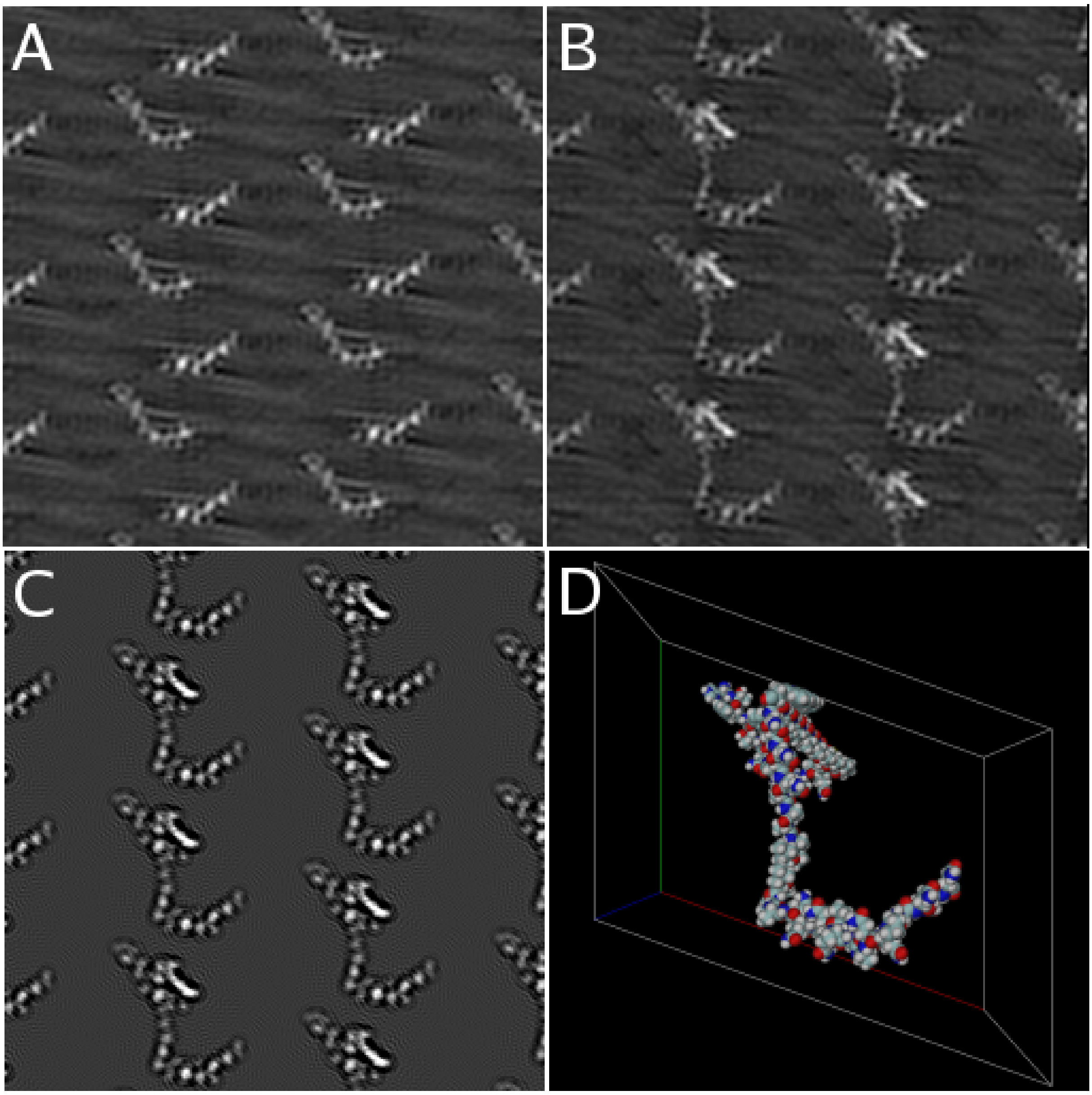


Figure 3.5: Reference images created by MAKETRAN and iplt. defocus=1000A,a=90, b=66, γ=110 and c=110. Cs=2 kV=200(A) Reference image generated by the unmodified MAKETRAN program (B) Reference image generated by a modified MAKETRAN omitting sync function interpolation for untilted images. (C) Reference image generated by an iplt script using the llpredict algorithm (D) 3D representation of the artificial protein generated by DINO.

the peak profile given to QUADSERCH and the cross correlation peak is taken as the masking criteria. To fill small holes and to reject small isolated regions in the mask, the mask is subjected to two rounds of filtering by consecutive dilation and erosion algorithms (Fig. 3.6). These morphological algorithm are implemented as part of a modified version of QUADSERCH. After masking the crystal, the unbending procedure is started once more with the masked image.

3.2.6 Projection Image Generation

Greyscale images of the averaged unit cell are generated in addition to the contour plots by backtransforming the projection data using FFT from the ccp4 program suite. The greyscale representation was chosen because it proved to be more intuitive for the novice user. For a better overview the image is generated containing 2×2 unit cells. After backtransforming, the image is sheared and scaled to account for the unit cell vector lengths and the unit cell angle. The image is subsequently annotated using the Python Imaging Library (`www.pythonware.com/products/pil`), adding the unit cell vectors, the tilt axis and icons for additionally applied rotations and the mode of unbending are added.

3.2.7 Phase Origin

Manual phase origin determination was implemented using the SGI utility XV to display the greyscale representation of the averaged unit cell. The new center of the unit cell can be interactively selected using the magnifying glass tool . The conversion of the pixel values - given for the center by xv - to the phase shift to be applied is carried out with the program MRCPHASE originally written by H. Stahlberg and modified by me to account for the different image dimensions in different projects.

Automatic determination of the phase origin can be done using the program ORIGTILT of the standard MRC program suite in case a reference dataset is available.

In AQP2 the phase shifts needed for merging the first untilted and lowly tilted images to generate a first 3D density were determined by eye.

In the SoPIP2;1 project phase origin determination for the first untilted and small tilt images was done using a script I wrote myself using an average of a single layered SoPIP2;1 crystal as reference. The image data was cross correlated with the reference and the phase origin was calculated from the cross correlation peak. Prior to merging, the phase origins were inspected manually to ensure a correct phase shift. A single

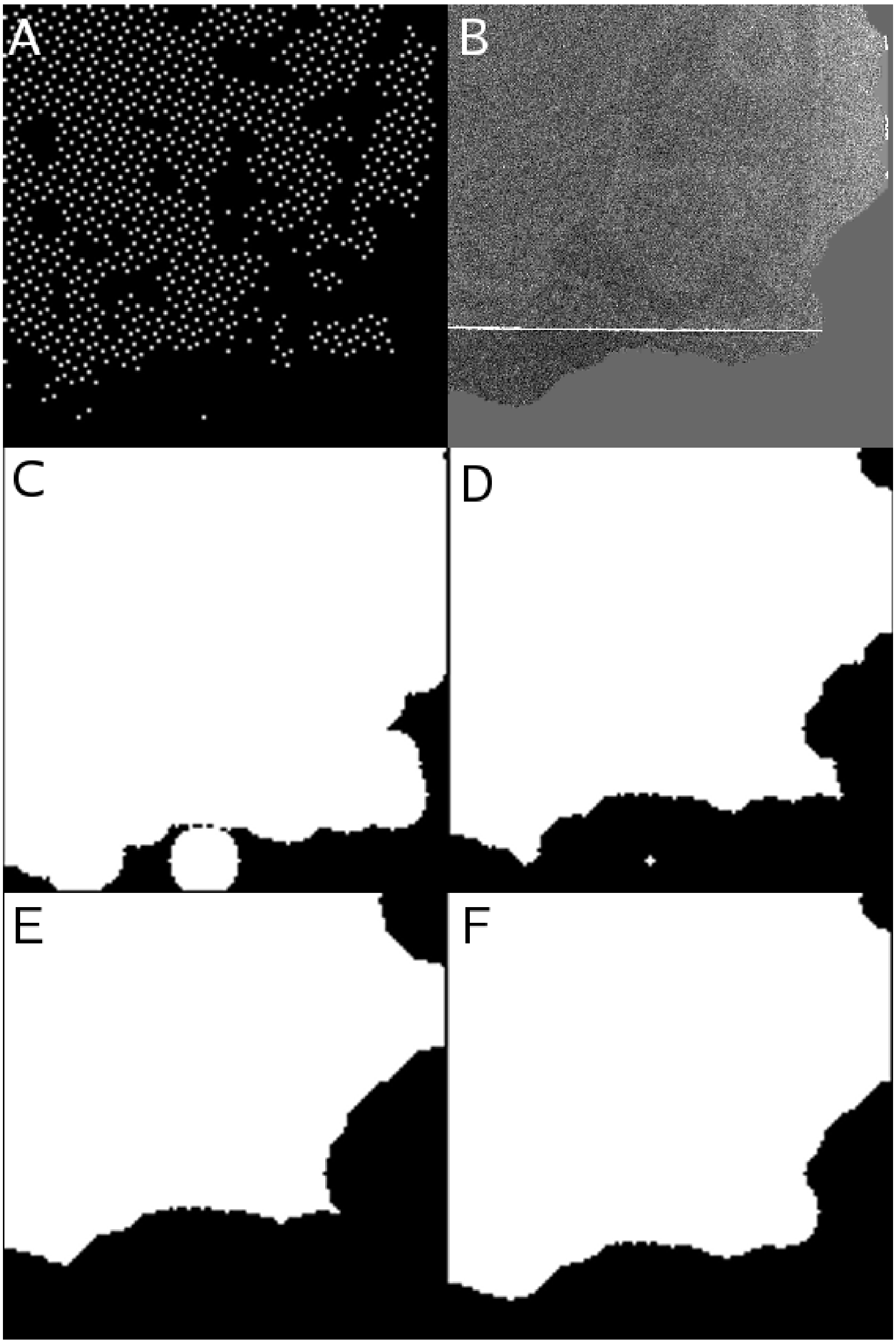


Figure 3.6: Mask generation by morphological algorithms. A mask for the image is generated based on the cross correlation image, which is then filtered by consecutive dilation and erosion algorithms.(A)cross correlation image(B)masked image(C) 1st dilation (D) 1st erosion (E) 2nd erosion (F) 2nd dilation

layer reference was taken to avoid wrong register shifts due to the still quite high cross correlation signal of the mirrored layer.

Recently, a more direct method to determine the phase origin has been implemented, avoiding interpolation in Fourier space. The structure factors of the to be shifted image are cross correlated with a slab generated from a reference map by the IPLT algorithm LLPREDICT (3.4.12), which uses an explicit Fourier summation to prevent interpolation. The optimal phase origin can then be directly calculated from the highest cross correlation peak.

3.2.8 Symmetrization in 2D

At the beginning of an image processing project it is often desirable to try different symmetries on an averaged dataset for visual inspection of the result, or to try out the different symmetries proposed by ALLSPACE. To use a full fledged merging and refinement tool as ORIGTILT for this task is overkill and to use the symmetrization tools of ccp4 is somewhat cumbersome because the symmetry axis definitions in ccp4 differ in some cases (e.g. $P22_12$) from the ones used in MRC. To this end, a small library was written in C++ to simplify the symmetrization in 2D. $P22_12$ symmetrization for AQP2 and the 92 Å crystals of SoPIP2;1 and p4 symmetrization for the 65 Å crystals of SoPIP2;1 were implemented. The symmetrization program reads a reflection data file in either hk,hkl or aph format and after symmetrization writes out a second reflection file in the same format. Additional symmetries can easily be added using the same library framework.

3.3 New Developments in Processing of Electron Diffraction Data

Even though MRC provides tools to process electron diffraction data, these turned out to be too inflexible for the processing of AQP2. For example the unmodified programs could only process images with a depth of 8 bit, therefore restricting the depth resolution of 16bit provided by the CCD. Additionally, the Unix version of the program SYNCFIT facilitating the discretization was restricted to the use of p3 symmetry.

As a consequence the processing of electron diffraction images was re-implemented in IPLT focusing on usability, flexibility and the possibility to automate processing tasks.

Each of the high level steps has been implemented as an individual python script, based on several low-level C++ modules.

3.3.1 Data Extraction

After indexing the diffraction pattern the amplitude of every peak has to be read out and corrected for the local background. Iplt implements two different extraction strategies, allowing cross checking of the results. The individual peaks can either be fitted by a gauss function or numerically integrated.

Gauss Fit: At the position of each putative diffraction spot, predicted by using the lattice, a two-dimensional gauss function is fitted:

$$f(x,y) = Ae^{\left(\frac{(x-U_x)\cos(\omega)-(y-U_y)\sin(\omega)}{B_x}\right)^2+\left(\frac{(x-U_x)\sin(\omega)+(y-U_y)\cos(\omega)}{B_y}\right)^2} + xP_x + yP_y + C$$

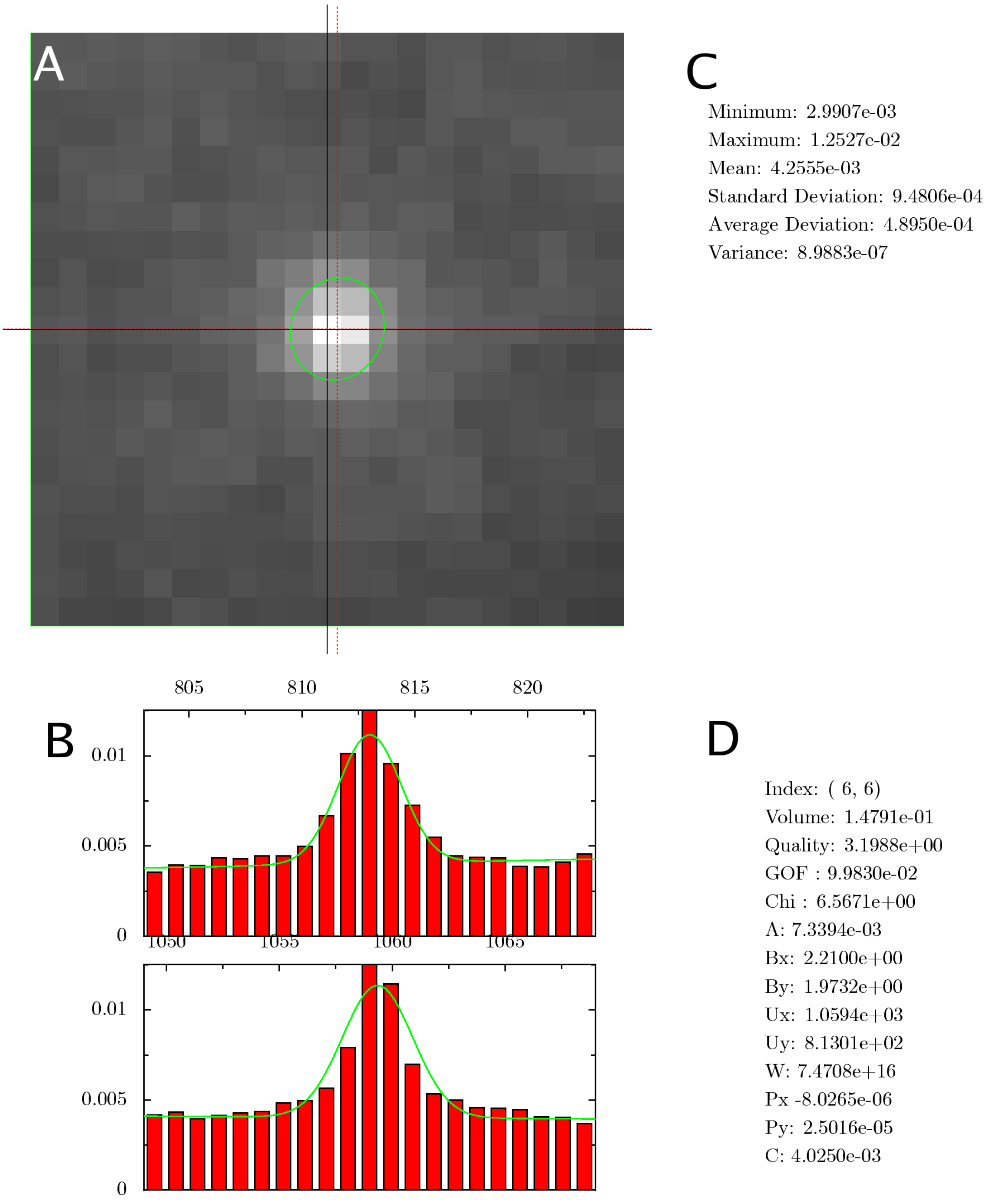
The gauss function consist of 9 parameters: peak height (A), half width in x and y (B_x, B_y), rotation angle (ω), position in x and y (U_x, U_y), and three parameters for a background plane (P_x,P_y,C) (see Fig. 3.7). Inclusion of a background plane proved to be crucial for a proper fit due to the high amount of background generated by inelastic scattered electrons.

Peak Integration: The peak integration algorithm integrates the diffraction peaks in concentric squares. The integrated peak volume is background corrected using the ring of pixels adjacent to the integrated square to calculate the mean background. Continuous increment of the square size gives peak volumes calculated from a increasing area (see Fig. 3.8). The algorithm stops at a given square size or if the background corrected peak volume difference between two cycles lies below a given threshold.

3.3.2 Origin Determination

During recording of a diffraction pattern the central beam is normally blocked by a beam stop to prevent damage to the scintillator of the CCD camera. Therefore, the center of the diffraction pattern, and with that also the origin of the diffraction lattice, is covered by the beam stop.

This necessitates the determination of the origin by indirect means instead of directly assigning it. For every potential origin, based on the predicted lattice point positions,



Minimum: 2.9907e-03
Maximum: 1.2527e-02
Mean: 4.2555e-03
Standard Deviation: 9.4806e-04
Average Deviation: 4.8950e-04
Variance: 8.9883e-07

Index: (6, 6)
Volume: 1.4791e-01
Quality: 3.1988e+00
GOF : 9.9830e-02
Chi : 6.5671e+00
A: 7.3394e-03
Bx: 2.2100e+00
By: 1.9732e+00
Ux: 1.0594e+03
Uy: 8.1301e+02
W: 7.4708e+16
Px -8.0265e-06
Py: 2.5016e-05
C: 4.0250e-03

Figure 3.7: Gauss fit of a diffraction peak. (A) The black cross shows the predicted peak position, the red cross the fitted one. (B) cross sections of (A) along the red lines. Red bars represent pixel values and the green line the fitted gauss function. (C) Statistical information for this sub-image (D) Parameters of the fitted gauss function

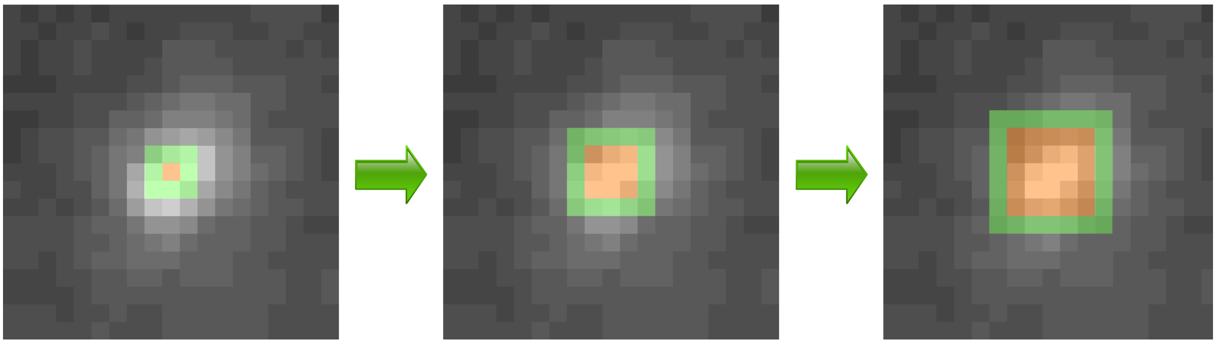


Figure 3.8: Peak integration. The pixel values are summed up in the red area giving the integrated peak volume. From the peak volume the average background is calculated as the mean value in the green adjacent ring is subtracted. Then the red area is expanded by one pixel in every direction. This procedure is repeated until the volume difference between two subsequent step is smaller than a given threshold.

the average amplitude difference between putative Friedel mates can be calculated. The correct origin is assumed to have the minimal average amplitude difference.

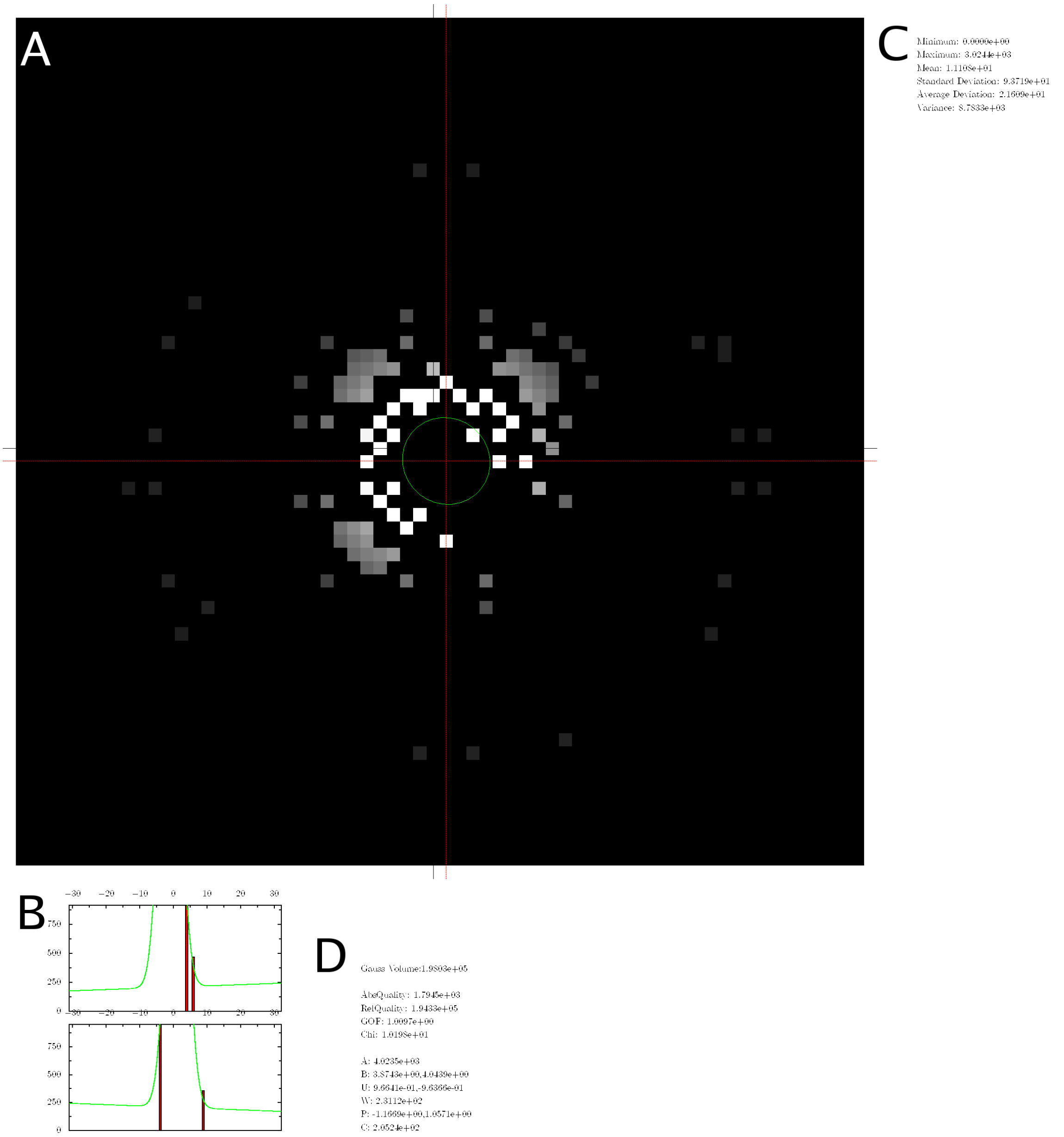
A second indication to where the origin lies is given by the background, caused by the inelastically scattered electrons. By fitting a 2D gauss function to the background values of the identified diffraction peaks, the origin can be found at the maximum of the gauss function (see Fig. 3.9). In this gauss fit, only backgrounds, determined by gauss fits with a given quality factor, are incorporated to avoid fitting of spots found beneath the beam stop. Both methods are implemented in IPLT to allow verification of the found origin position.

Because both of the methods are sensitive to incorrectly fitted peaks in the beam stop area, prior identification of the beam stop by a segmentation algorithm could further improve the reliability of the given methods.

3.4 General Developments Concerning Image and Diffraction Processing

3.4.1 Increased Greyscale Depth

The scanner in our lab scans images in negative transmission mode at a color depth of 16bit (see 4.2.3 for details), and therefore all MRC programs were modified to allow image processing at this depth. This includes modification of the program LABEL to correctly convert a 16 bit negative transmission tif image to an MRC image with grey



```
Minimum: 0.0000e+00
Maximum: 3.0244e+03
Mean: 1.1108e+01
Standard Deviation: 9.3719e+01
Average Deviation: 2.1609e+01
Variance: 8.7833e+03
```

```
Gauss Volume:1.9803e+05

AbsQuality: 1.7945e+03
RelQuality: 1.9433e+05
GOF: 1.0097e+00
Chi: 1.0198e+01

A: 4.0235e+03
B: 3.8743e+00,4.0439e+00
U: 9.6641e-01,-9.6366e-01
W: 2.3112e+02
P: -1.1669e+00,1.0571e+00
C: 2.0524e+02
```

Figure 3.9: Explicit gauss fit of the background. (A) The black cross shows the predicted peak position, the red cross the fitted one. (B) cross sections of (A) along the red lines. Red bars represent pixel values and the green line the fitted gauss function.(C) Statistical information for this sub-image (D) Parameters of the fitted gauss function

values corresponding to the optical density (OD). In addition FORMAT statements and variable storage depths had to be changed in several programs to allow for grey values above 255 and with this also for summed amplitudes far above 65535. The format of the column based reflection files also had to be changed to give the amplitude column space for more digits.

3.4.2 Linux Port

One bottleneck during the processing of the AQP2 project proved to be the CPU speed of the used SGI Origin servers. To harvest the much increased processing power of modern CPUs the complete image processing setup was ported to Linux. A great help during porting was the fact that the image2000 version of the MRC programs was already compatible with Linux, therefore the changes contained in our local, heavily modified version of MRC had to be included in the image2000 version of the programs and only the self written programs had to be ported completely. Within the scope of porting, some modifications were also incorporated into the compiling and building stage of the MRC suite.

Version 2000 of the Image Library: The image2000 version of the MRC programs was built upon an image library containing a new header format for MRC images. The new header includes a machine stamp to discriminate big endian and small endian machines. To this image library I added a mechanism for recognizing the byte order also for old style MRC images, providing therefore a more seamless migration from the SGI big endian architecture to a small endian Linux architecture.

MA21AD: The Harwell subroutine MA21AD was unfortunately not compatible with the Intel Linux compiler when used in a dynamic library and was therefore replaced by a compatible q-r decomposition[36].

Dynamic MRC Library: In the original MRC distribution, executables are all statically linked to their corresponding libraries which leads to an increase in program size and also to an increased memory usage should several executables be run in parallel. As a second major drawback every change in a library necessitates the recompilation of all corresponding executables, because the build script does not separate the compile and link stage. My modified version of MRC consist of one single shared library containing all the subroutines used by more than one executable. This also includes the subroutines which are duplicated in the original version of MRC, like for example

ASYM which was present in TTREFINE,ORIGTILT,MAKETRAN and MERGEDIFF. All the executables are dynamically linked to the MRC library; in consequence, relinking is only necessary in case of modifications in the library.

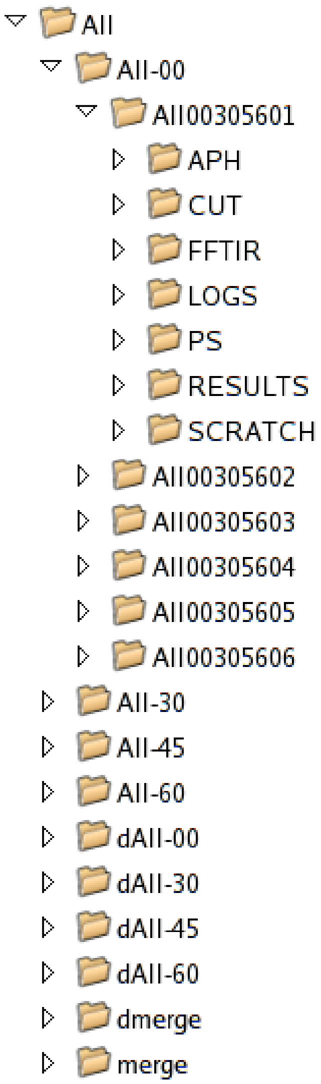
Scons Build System: The build scripts shipped with MRC were replaced by the scons build system also used in IPLT. The scons build system allows easier integration of new programs and provides the feature of recompiling and relinking the modified part of a project instead of recompiling all executables once a modification is made. The scons build system includes also a platform identification and is consequently capable of finding the corresponding compiler and link flags for the current platform automatically. This renders the maintenance of individual build scripts for different platforms unnecessary.

SVN: The versioning system svn (`subversion.tigris.org`) was introduced to keep track of all changes made to MRC sources and the self written software.

Fastest Fourier Transform in the West (FFTW): The MRC contained FFT subroutine was replaced by a version calling the FFTW[10] library. This proved to speed up calculation of Fourier transforms considerably.

3.4.3 Standardized Directory Structure

A standardized directory structure was established for all the projects in the group. This simplified data retrieval and directory traversal for scripts working on several image directories (e.g. the directory walker (see 3.4.4)). The main directory of the project contains a directory for every tilt angle set of either images or diffraction patterns and a directory for every merge. Within these directories, a subdirectory is created for every image. The image directory holds folders for all reflection files (APH),for the subimages used to determine the defocus for tilt axis determination (CUT), for all Fourier transforms (FFTIR), for all logfiles (LOGS), for the postscript output (PS), for the postscript output of previous processing runs (RESULTS) and for all temporary files (SCRATCH). Figure 3.10 shows part of the directory tree of the AQP2 project.



```
All
  All-00
    All00305601
      APH
      CUT
      FFTIR
      LOGS
      PS
      RESULTS
      SCRATCH
    All00305602
    All00305603
    All00305604
    All00305605
    All00305606
  All-30
  All-45
  All-60
  dAll-00
  dAll-30
  dAll-45
  dAll-60
  dmerge
  merge
```

Figure 3.10: Standard directory structure. The shown tree is part of the directory tree of the AQP2 project. Only six image directories originating from one negative are displayed.

3.4.4 Directory Walker

To change parameters in multiple meta-info files or to start a refinement in several files, I implemented a flexible directory walker which recursively traverses all directories in a project and starts a given task in the directories matching a pattern given to the walker. The directory walker is written in python using a generator function in combination with the OS.WALK class.

3.4.5 Load Balancing

A load balancing mechanism similar to the one included in the script responsible for image splitting and preparation (see 3.2.2 for details) is also available as a wrapper script for a user defined processing task. The load balancing is implemented in a way that the given task is only started if the average load of the machine for a given time period is below a user set threshold. For example, the user is able to start a refinement task for several images in the evening using the given wrapper script (wittingly named nightjob) to take advantage of idle CPU cycles during night.

3.4.6 Storage Handling

Several of the projects I was involved in used disk space of several hundred Gigabytes and corresponded to processing times of several months to some years. Having a backup concept to prevent data loss in the case of a hardware failure is obviously crucial. To achieve this I set up a file server with two Transtec storage RAID[2] arrays working in RAID5 mode to store the project data. Having the project on a RAID array already prevents data loss in the case of single hard disk failure. To further secure the data additional backup strategies were implemented.

To avoid loosing project data in the case of a complete RAID failure, the raw image data together with the meta-info data needing for reproducing the achieved results is additionally mirrored on a second available RAID. In the case of a complete RAID failure all the projects data should therefore still be available, the project just needs to be reprocessed using the stored meta-info data. In the case of a fire in the computer room or a water leak both RAIDs could still be damaged leading to the complete loss all of the projects in the group. To avoid this worst case scenario a backup of the essential image data to a server in an other building is in planning.

---

[2] Redundant Array of Inexpensive Disks

In addition to the image data, the meta-data can be backuped individually for a project. Different refinement stages can be backuped and also be rolled back in the case of an badly parametrized automatic refinement wrecking havoc or a user accidentally entering wrong parameters. Every backup set can be named individually and retrieved using the given name.

3.4.7 Image Display with Ximdisp

The functionality of the Image display program XIMDISP included in the MRC software package was expanded to speed up recurring task during image processing.

Default minimum and maximum grey values used for displaying images: The average minimum and maximum value used for scaling and displaying an image in XIMDISP seemed to be more or less constant for a given project recorded on one microscope. Therefore, for convenience, the default minimum and maximum values used for displaying images and diffraction patterns are saved in a config file in a users home directory, which removes the necessity for the user working on one project to enter the values for every displayed image manually.

Reading and Writing of the Meta-Info File: Easy and straightforward ways to read data from and write data to the meta info file were implemented. If a meta-info file is found in the current working directory, XIMDISP reads it upon start and with this extracts the images name, masking status, defocus parameters and already indexed lattices. The user can now choose from a menu if he wants to read the original image (masked or unmasked), the Fourier transform of the image, a reduced Fourier transform, or a Fourier transform of the central section of an image to fit the defocus values.

Defocus: For images with a very low contrast, where the contrast transfer function (CTF) cannot be fitted automatically, the defocus can be manually adjusted in our modified version of XIMDISP. Upon entering the CTF fitting menu the defocus and astigmatism is read from the meta-info file. The contrast transfer function $T(k)$– k being the spatial frequency– can be calculated using the defocus value Δf, the amount of amplitude contrast C_a, and the spherical aberration C_s using following formula[38]:

$$T(k) = -\sqrt{1-(C_a)^2}sin(\phi(k)) + C_a cos(\phi(k)) \tag{3.1}$$

$$\phi(k) = \frac{\pi}{2}(C_s\lambda^3|k|^4 - 2\Delta f(k)\lambda|k|^2) \tag{3.2}$$

The Thon rings, which correspond to the zero crossings of the CTF, are displayed as green ellipses overlaid over the Fourier transform of the image (Fig. 3.11) and the defocus and astigmatism can be changed interactively. Once the fit of the overlaid CTF and the Thon-Rings seen in the image is sufficient, the modified defocus values can be written back to the meta-info file.

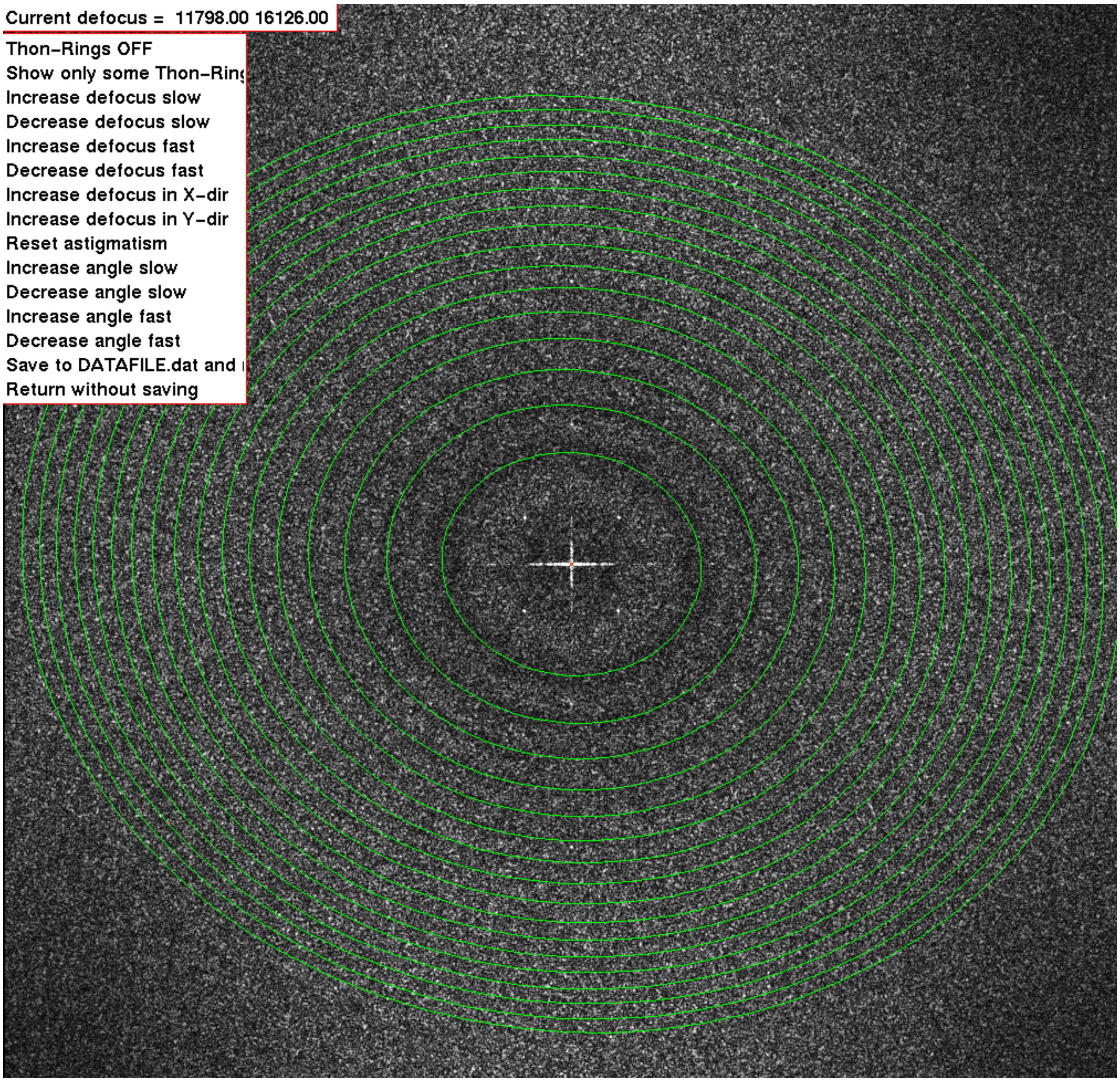
Lattice Determination: The manual lattice indexing in XIMDISP was extended to include an index prediction for newly clicked spots once an initial lattice had been determined (see Fig. 3.12). This gives the user the possibility to refine a lattice without the tedious work of entering the lattice indices for every newly clicked spot. The lattice determination menu also allows to read or write two lattices to the meta-info file (see Fig. 3.13), which is used in the case of an epitaxial twinned lattice.

Spotlist: Once the lattice (or the lattices in the case of epitaxial twinned crystals) are determined, a list of good spots has to be prepared which will then be used in the masking step during unbending. Eligible spots can be determined by eye. To this end, XIMDISP was extended to produce a user defined list of spots. If a lattice is displayed, a subset of the spots can be selected with the mouse, and the indexes of these spots can be written to a file (see Fig. 3.14). (The original way of doing the same task would have been to count the indices of the good spots manually on the screen and then preparing a text file containing the indices with a text editor.)

3.4.8 Lattice Determination

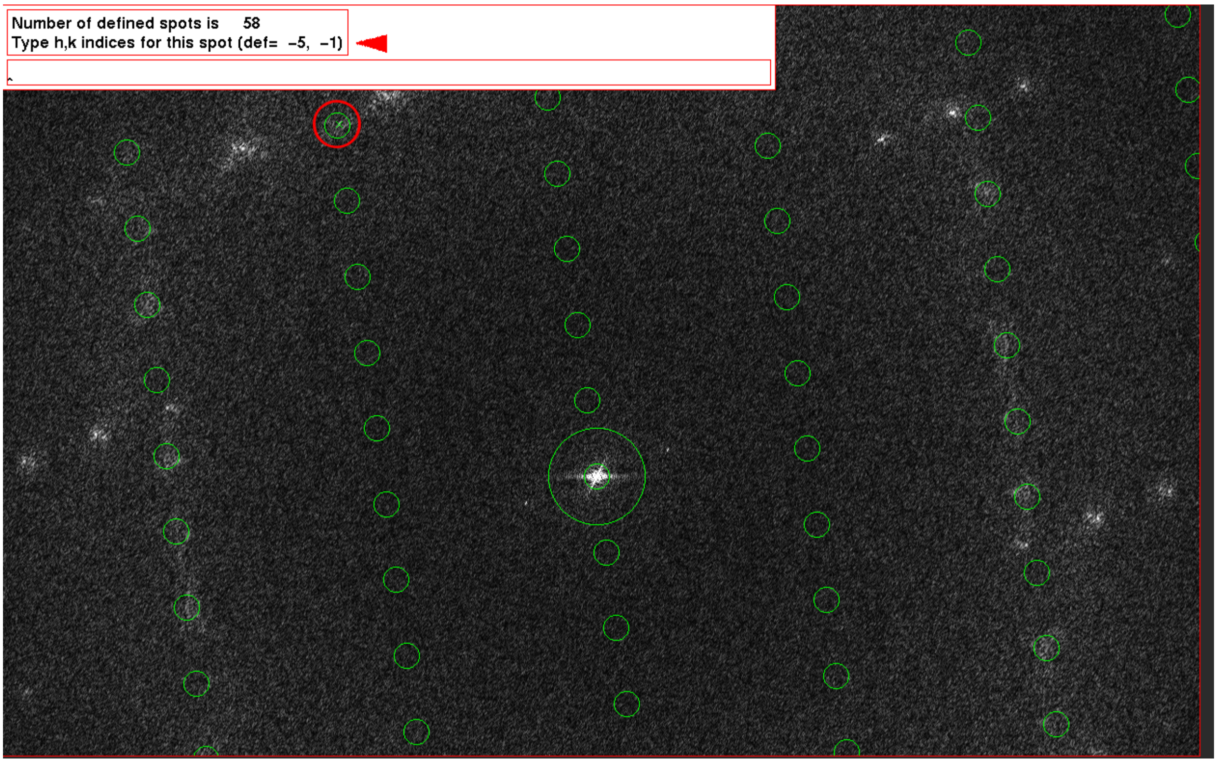
Automatic determination of a lattice normally includes two steps. First, a number of candidate peaks possibly lying on the to be indexed lattice have to be found. Second, from the found peaks the optimal lattice has to be calculated.

Rotational Search of Known Lattice: If the length and the angle between the two unit cell vectors is known, the lattice can be rotated around the origin until a maximum number of peaks from the previous peak search fit onto the lattice. During rotation the lattice has to be distorted according to the tilt geometry. The implementation of this algorithm was done in the standalone program MRCFINDLAT developed by H. Stahlberg and myself. The approach is quite robust and insensitive to noise. The main drawback of this algorithm is the required previous knowledge: I) the size of



Current defocus = 11798.00 16126.00

Figure 3.11: Ximdisp showing an overlay for interactive CTF fitting. In the top left corner the current defocus values are displayed. The menu on the left allows interactive adjustment of the amount of defocus in x and y direction and the rotational angle.



Number of defined spots is 58
Type h,k indices for this spot (def= -5, -1)

Figure 3.12: Index hinting. As soon as a first lattice is indexed or read from the meta info file, a hint is displayed (red arrow) for every newly clicked spot (red circle).

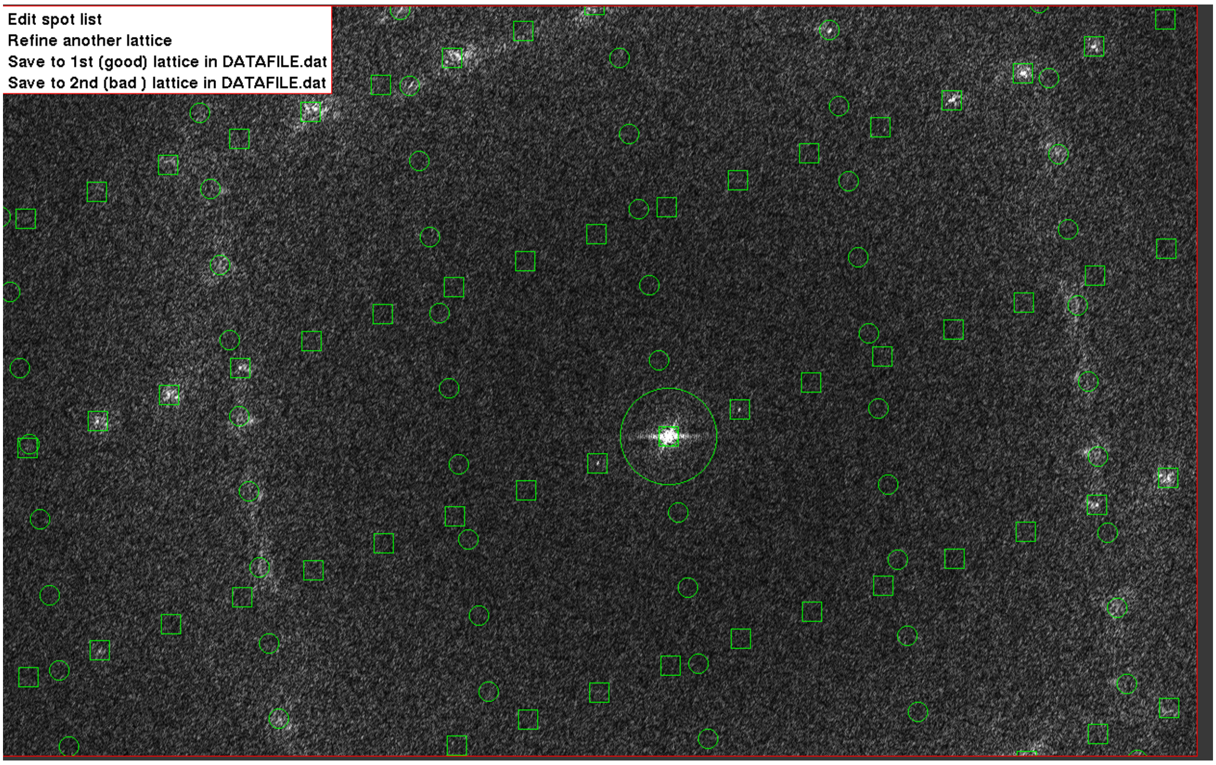
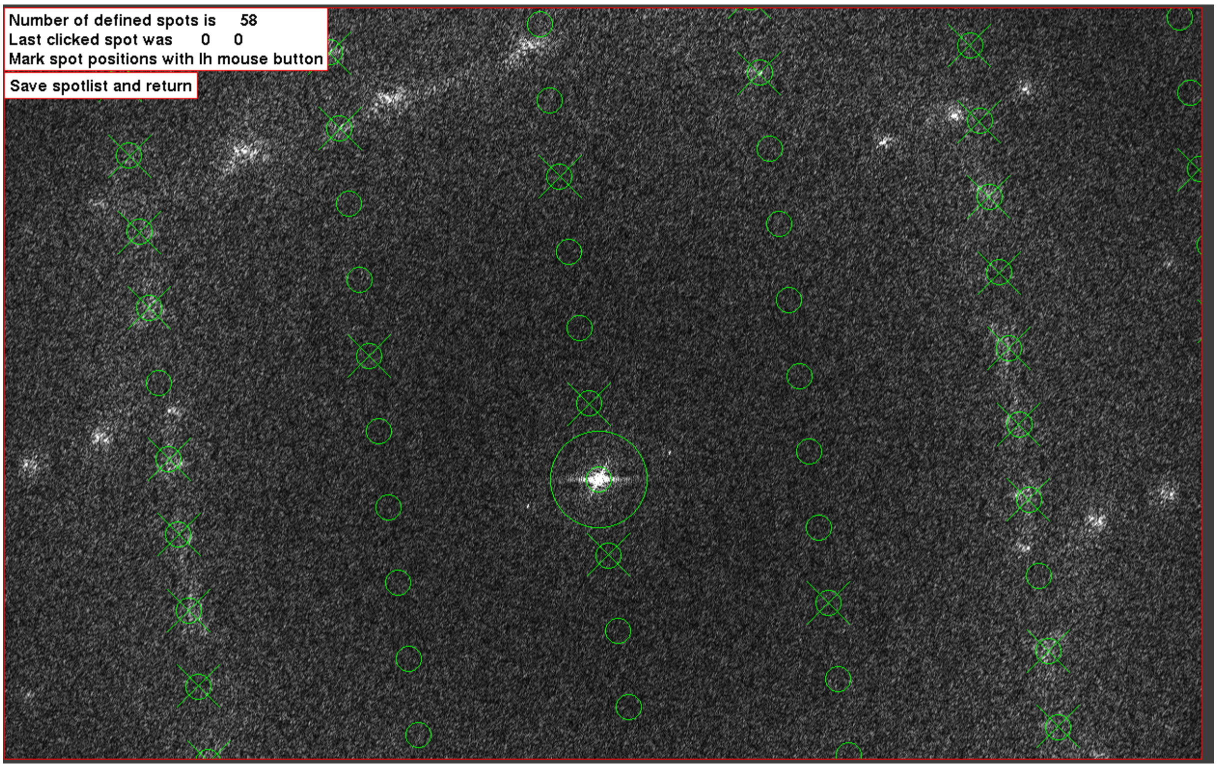

Edit spot list
Refine another lattice
Save to 1st (good) lattice in DATAFILE.dat
Save to 2nd (bad) lattice in DATAFILE.dat

Figure 3.13: Storing of an indexed lattice to a meta file. The example shows the two indexed lattices for a KdgM epitaxial twinned crystal. The lattices are represented by green circles and squares respectively. The big green circle marks the lattice origin. In the top left corner the menu can be seen with which the lattice can be saved to a meta data file.



Number of defined spots is 58
Last clicked spot was 0 0
Mark spot positions with lh mouse button

Figure 3.14: Spotlist generation. Individual spots represented by green crosses can selected and deselected using the mouse.

the unit cell vectors have to be known quite precisely. II) If the image was recorded on a tilted sample, the tilt geometry has to be known to calculate the lattice distortion. In this case the tilt geometry has to be determined by using the defocus gradient in the image, because otherwise the lattice determination would rely on the tilt geometry determination and the tilt geometry determination using Shaw's algorithm[40] would rely on the lattice determination, leading to a chicken and egg problem. III) The origin of the lattice has to be known beforehand, consequently this algorithm is not applicable to diffraction patterns.

Difference Vectors: The second approach recently implemented in IPLT by our group calculates the difference vectors between all pairs of peaks found during peak search. These difference vectors are then summed up in a second image. On this image a second peak search is performed and the two peaks found closest to the origin which are not collinear yield the two unit cell vectors. For the case $\left|\vec{a}+\vec{b}\right| < \max\left(|\vec{a}|, \left|\vec{b}\right|\right)$ or $\left|\vec{a}-\vec{b}\right| < \max\left(|\vec{a}|, \left|\vec{b}\right|\right)$,which is possible for highly tilted images or special symmetries, this algorithm will in principle find the correct lattice but the indexing will be wrong (see Fig. 3.15). In this case it is indispensable to have some prior knowledge of the lattice vectors or the approximate tilt geometry.

3.4.9 Reindexing

A problem occurred during image processing for square unit cells that did not exhibit P4 symmetry, and therefore the two unit cell vectors could not be exchanged. Namely the double layered AQP2 crystals have a $p22_12$ symmetry and the (rarely found) SoPIP2;1 crystals with a unit cell of a=b=92 Å and a $p22_12$ symmetry showed this kind of difficulty. Because the two unit cell vectors could not be discriminated during the indexing step, the averaged image data of each image had to be compared to a reference prior to merging to find the correct orientation.

Images were compared to the reference using a extended version of ORIGTILT, which was modified to carry out a reindexing on the fly for the case that a special flag was set. By comparison with a reference, which was originally created using images where the correct orientation could be determined by eye, the correct orientation of newly processed images could be found.

For diffraction patterns a similar procedure was implemented using the LLPREDICT algorithm for reference generation in iplt.

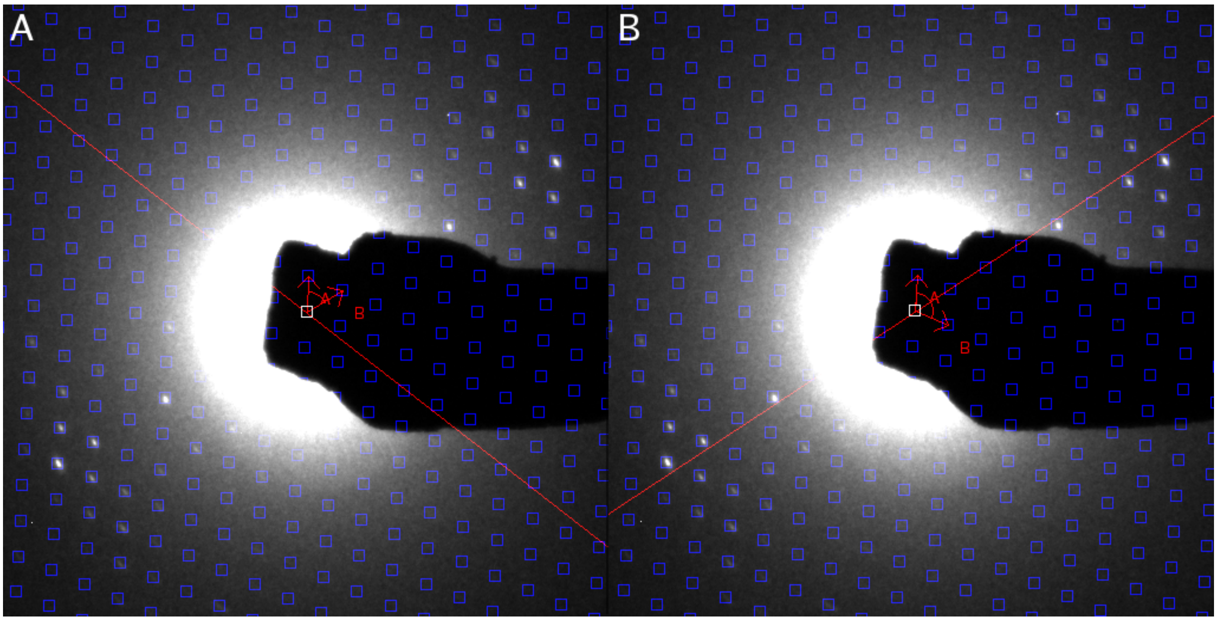


Figure 3.15: The iplt lattice viewer displaying displaying wrong and correct indexing of a tilted diffraction pattern. The blue squares show the indexed lattice. The two red arrows represent the reciprocal unit cell vectors and the long red line indicates the tilt axis. (A) Wrong indexed lattice determined by the difference vector method. (B) Manually corrected lattice showing the expected tilt axis.

3.4.10 Merging

Speed Improvements: One bottleneck during the processing of the AQP2 data was found to be the step of merging the several hundred images to one 3D dataset. Profiling of ORIGTILT lead to the astonishing result that most of the processing time was not consumed by the merging itself, but by the sorting of the reflections prior to writing them to disk. By replacing the existing bubble sort (which scales with a complexity of $\Theta(N^2)$) by a quicksort algorithm (which scales with a complexity of $\Theta(N \log N)$) I achieved a 5 fold increase in processing speed for the merging step.

In addition to the improved sort algorithm, the merging was separated by a newly created script into different threads, merging subsets of the data to be merged individually, and then in the end merging the subsets. This makes it possible to harvest the full processing power of a multiprozessor system.

Cyclical Refinement: To refine an initial 3D mass density, a cyclical refinement procedure was implemented using the program ORIGTILT for merging and also for the determination of refined image parameters like tilt axis, tilt angle, beam tilt and phase origin. In contrast to most of the refinement schemes used in MRC, the refinement developed in our group uses discretized dataset as reference which was fitted by LATLINE. With this setup the additional symmetry constraints in the AQP2 double layers, which are enforced by latline (see 3.4.11), can be exploited.

3.4.11 Discretization

The packing arrangement in double layered AQP2 crystals consisting of 2 p4 symmetric layers gives rise to additional phase restriction along lattice lines with $h = k$ or $h = -k$ not present in a normal $p22_12$ symmetry. To optimize the obtained real space dataset, LATLINE was modified to take these additional phase restrictions into account, during the fitting step.

3.4.12 Lattice Line Prediction

A recurring difficulty during processing of EM data is the continuity of the data along the z-axis. Unlike the data processed in X-ray crystallography, a 3 dimensional dataset in 2D electron crystallography is continuous along the reciprocal axis perpendicular to the crystal plane, usually referred to as z*, due to the fact that a protein lipid array is not crystalline in this direction.

In order to utilize tools written by the X-ray community, the dataset has to be discretized at one point. If this discretized dataset is then compared, at a later stage, to a central section, the data points needed along a lattice line normally don't coincide with the data point provided by the discretized dataset.

To solve this problem, a novel algorithm was implemented in our group to predict amplitude and phase at a arbitrary position along a given lattice line. The algorithm takes a real space dataset as a reference, which is then Fourier transformed by a FFT algorithm in x- and y-direction, and pre buffered. An explicit Fourier summation is then used to calculate the lattice line value at a given z*.

3.4.13 Projects Statistics and Consistency Tests

To check the consistency of the processed data, the project subdirectories are traversed using the directory walker (3.4.4) and some basic tests on the meta-data accumulated during image processing are performed. Some tests are done on the single image level and other tests include all the images originating from the same negative. For all images where at least on test fails, an entry is recorded in a log file and the image will be inspected manually. This test framework was inspired by the different unit test frameworks commonly used in modern programming paradigms.

Following tests are performed on the single image level:

1. The angle between the x-axis and the tilt axis (axis angle) should be constant within some range for a given microscope setup. For images recorded on the CM200FEG at a magnification of 50'000×, the axis angle was determined to be roughly 60°. This was determined by averaging the axis angle of all tilted AQP2 images, excluding clear outliers. If the axis angle of a tested image deviates more than 15° from the nominal axis angle, the test is flagged as failed. The plotted axis angles for all tilted images of the AQP2 project are plotted in Figure 3.17.
2. For passing the second test the measured tilt angle must lie within a range of ±5° to the nominal tilt angle.An example for a tilt angle plot can be found in Figure 3.16.
3. The lattice is tested to be right handed.
4. A final test is performed to make sure that the phase origin shift needed for merging is set.

All groups of images originating from the same negative are tested for the following conditions:

1. The image group is tested for the consistency of the different image processing flags like 90° rotation in AQP2, the flag to treat an image as untilted or the flag marking a spot scan image.
2. Image groups where the axis angle between the individual subimages deviates more than 5° are flagged as erroneous.
3. If the tilt angle within a group deviates more than 5° the group is logged as possibly failed.
4. To pass the last test, reciprocal unit cell vectors of all the subimages have to lie within a distance of 5 pixel.

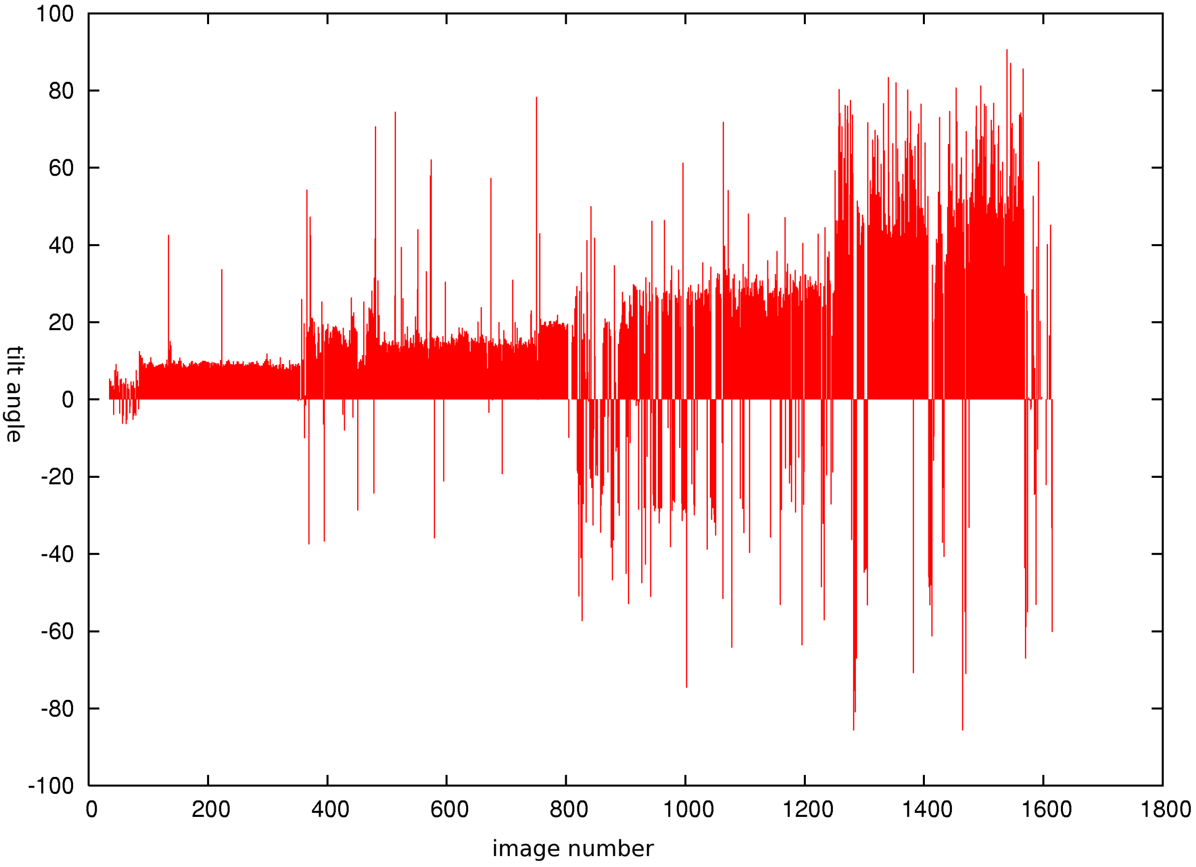


Figure 3.16: Tilt angle plot. Every bar in the plot corresponds to one tilted AQP2 image. The images are sorted by image name and therefore also by nominal tilt angle. By plotting the tilt angle several outliers could be identified which had to be corrected manually. Similar plots were also plotted for SoPIP2;1.

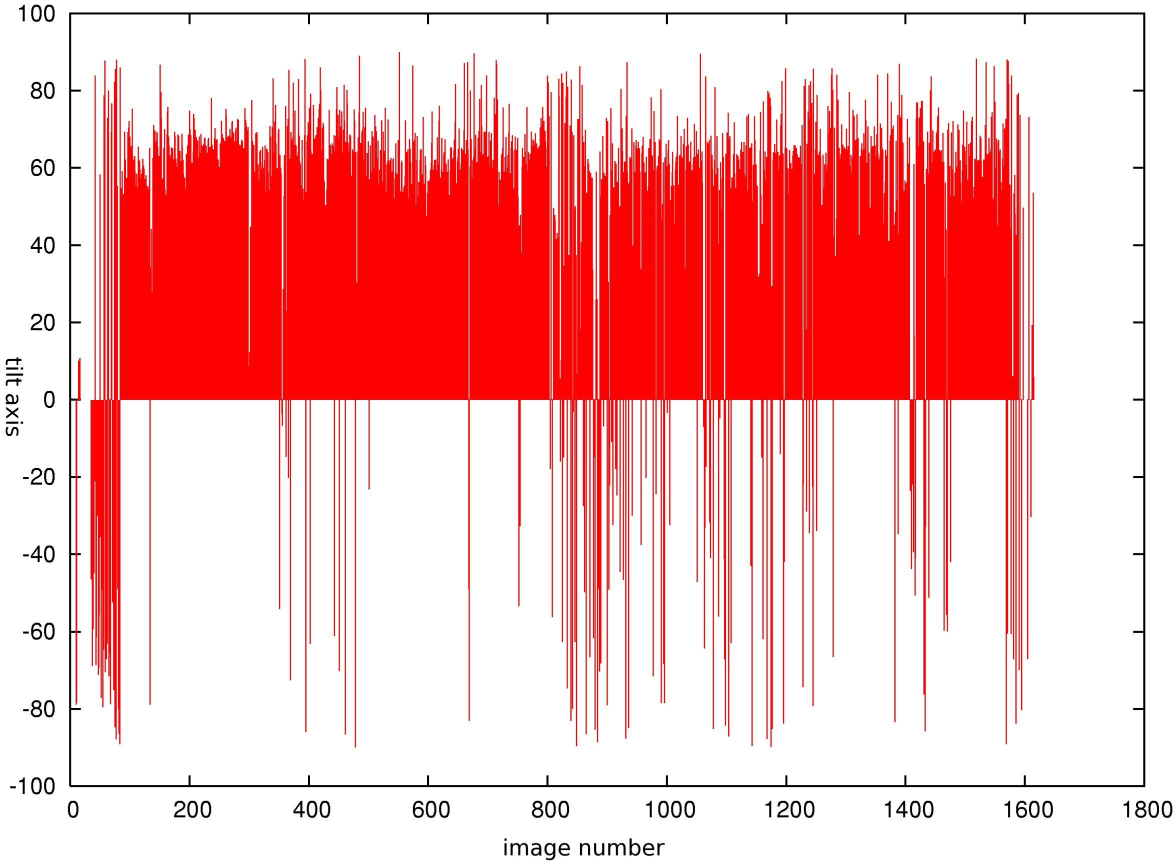


Figure 3.17: Axis angle plot. Every bar in the plot corresponds to one tilted AQP2 image. The images are sorted by image name and therefore also by nominal tilt angle. The expected axis angle for the tilted images is 60°. Especially for only slightly tilted images several images having big deviations in the axis angle could be found. Similar plots were also plotted for SoPIP2;1.

3.4.14 Interoperability

Users processing electron microscopy images or electron microscopy diffraction patterns are often confronted with the fact that they have to learn different interfaces and different scripting languages to use the wide variety of software available. Usage of different software packages is unavoidable, because none of the existing packages provides all the functionality needed for processing electron microscopy data to a level to achieve an atomic structure of a membrane protein. Therefore, the user must use several packages with different interfaces and somehow either glue them together with shell scripts or start the individual programs by hand leading to a low throughput of images and in general a very cumbersome work flow.

To counteract these problems a python wrapper for the most common image processing packages used in 2D electron crystallography (MRC,CCP4,SPIDER) has been written to provide an interface conforming to IPLT (see Fig.3.18). As a result, the user can now write his complete image processing scripts in the powerful scripting language PYTHON and mix different algorithm using the different processing packages in a transparent way. Interactive working on a python prompt is also possible.

The python wrapper itself consist of four parts: a common part responsible for disk space allocation, an image handle encapsulating the image data, an reflection handle encapsulating the reflection data text files and an algorithm class encapsulating the individual executables of mrc, ccp4 spider and the self written work.

Image Handle: The actual image was encapsulated in an image handle. In contrast to IPLT the MRC image handle still holds the image data on disk in a temporary image[3]. The image handle contains a APPLY method which can be used to apply an algorithm and consequently start the underlying executable with the image as the input image. The image created by the algorithm is then attached to the image handle and the old image is deleted if it was a temporary image. If the old image was explicitly loaded using the handlers LOAD method it is not deleted.

Reflection Data Handle: The handle for reflection data wraps all forms of column based text files, storing reflection data, in MRC. Upon loading a file, the corresponding reflection data is read into memory and subsequent operations like reindexing, phase

---

[3] I made an effort to adjust the image reading and writing routines of mrc to use a named pipe, which would have had the consequence that the image data could be held in memory, avoiding time intensive disk access, but unfortunately, the mrc routines use seek operations, which are not compatible with the named pipe mechanism.

shifts or merging of an image and a diffraction dataset are carried out in memory. Akin to the image handle the reflection data handle also has an APPLY method which can be used together with algorithms suitable for reflection data. By loading a reflection file into a reflection handle and re-saving it using a different format, reflection file format conversion can be achieved. Additional reflection file formats can easily be added by adding the column number and assignment of the new format to a config file.

Algorithms: Algorithm objects were created encapsulating the individual executables. Having the executables wrapped as algorithm with coherent interface allows seamless integration of the MRC and ccp4 routines into IPLT. For example, it is possible to generate a reference for unbending in IPLT using the LLPREDICT algorithm, and using this reference in the same script for the unbending algorithms in MRC. A default value is provided for every needed algorithm parameter within the algorithm so that the user only needs to enter data deviating from the default values. In addition, the implemented algorithm automatically takes image parameters from the provided meta-info file.

The algorithm class also contains a logging facility, giving the user the possibility to record the exact date and time an algorithm is executed on an image, resulting in a complete history of algorithms called and modifications done on an image.

Disk Space Allocation: The wrapper also contains a concept for allocation of temporary disk space. Every algorithm can reserve the approximately used disk space from a pool of scratch locations prior to execution of the underlying program. After execution, the reserved scratch space is released for use by other algorithms.

3.4.15 Meta Info

During processing of images and diffraction data a fair amount of meta data has to be saved and organised. In a first version, a text data file holding all the meta data for one image was introduced by H. Stahlberg. In this text file the first 40 characters of every line corresponded to one image parameters, and the second 40 characters of every line included a short description of the parameter. Although this design considerably improved meta data storage, there were several drawbacks:

1. Users changing parameters in the text file had to take care not to violate the fixed line length.

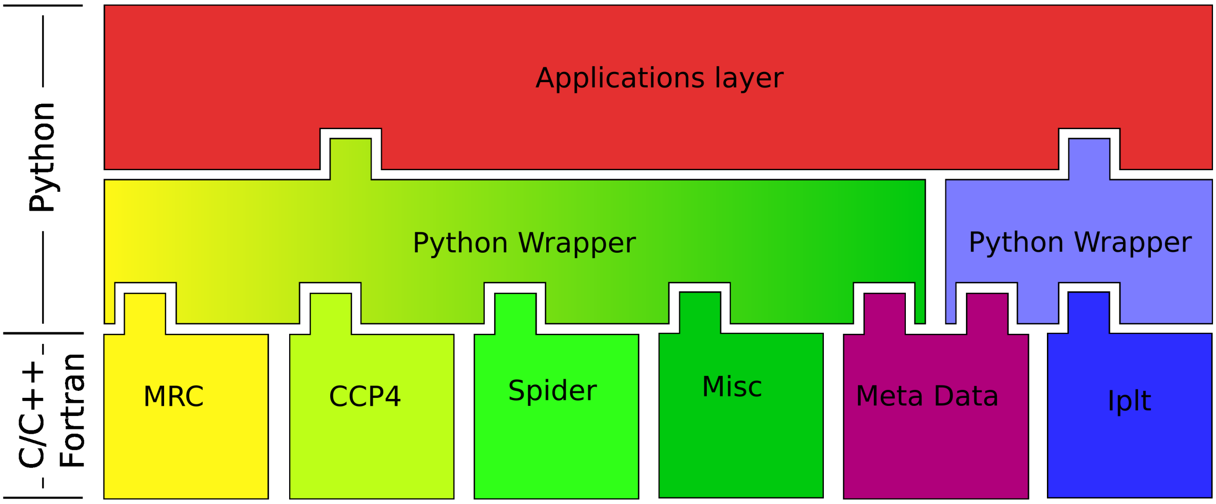


Figure 3.18: Building blocks of the image processing framework. Iplt by design already incorporates a python wrapper. For mrc, ccp4, spider and the self written software a python wrapper was additionally added to facilitate interplay with iplt. The meta data stored as XML data using a python interface to extract and modify data from within the python scripts of the wrapper and application level.

2. User defined remarks could not be added.

3. Because ever text file also contained data common to the overall project a lot of data was duplicated which proved to be error prone and cumbersome in the case a global parameter had to be changed.

4. In the original design every time a parameter was requested in a script, the text file was opened, the line containing the parameter was parsed, and the file was closed. Therefore, within a script, the text file was read several times, leading to a severe slowdown in parameter retrieval.

To address the last point I wrote a python class encapsulating the data file, which opens the file at the start of a script and from then on the provided parameters are from memory.

At a later stage, a meta info class using an XML representation for storing the data was introduced to the MRC python framework, providing a more flexible handling of meta data. The design of this new meta info class was inspired by the work originally started by A. Philippsen for IPLT, allowing an easy exchange of meta data with IPLT (Fig. 3.18). The XML files are organised in a hierarchical fashion (Fig. 3.19) to prevent data duplication. If, during parameter retrieval, a value cannot be found in the XML

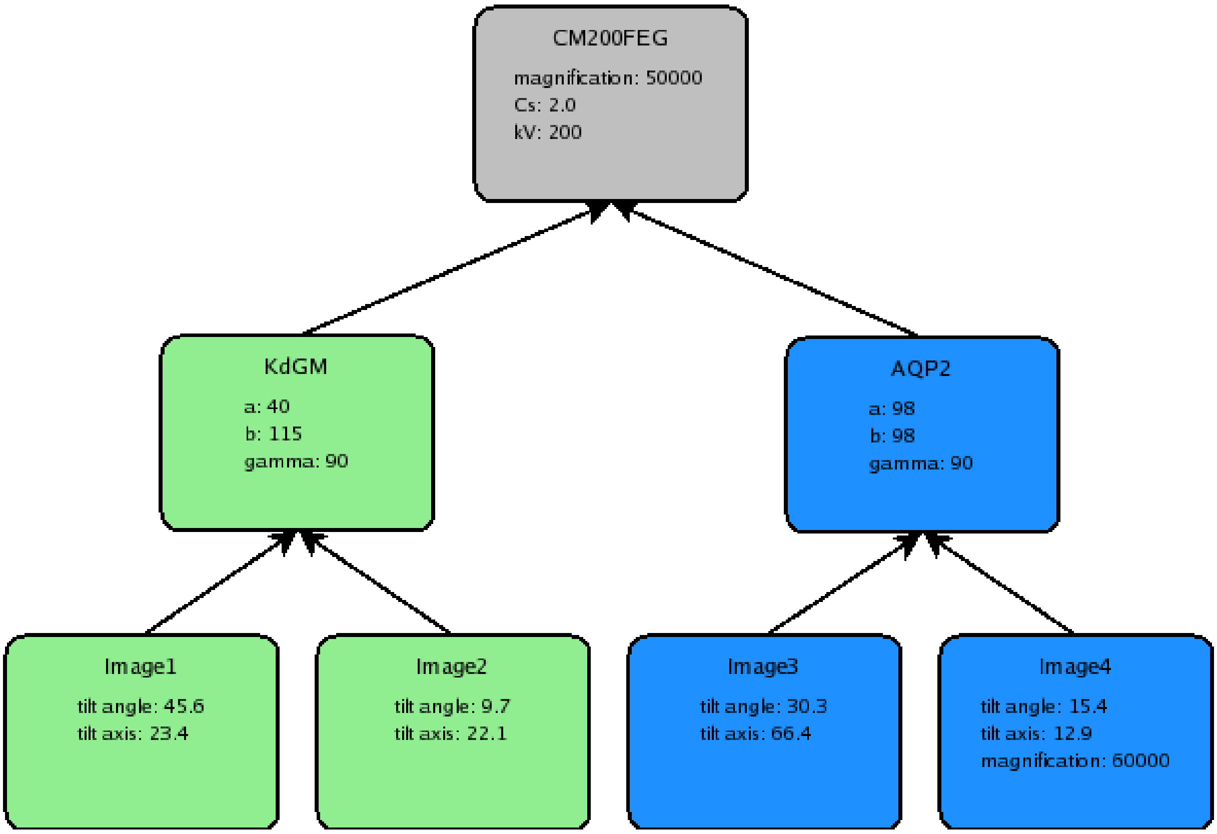


Figure 3.19: Hierarchical XML structure. An example for two projects using the same microscope is shown. The microscope parameters and default magnification are grouped together in one file and are automatically propagated to the meta-info of all image files. Image4 shows an example where the default magnification is overwritten.

file corresponding to the image, the search is recursively extended to the current XML's parent files.

An important feature of the data retrieval is the automatic conversion according to a data's type attribute. There are for different basic types implemented: integer, float, string and boolean. The strict typing in the data file simplifies identification of falsely entered data immensely and provides users writing scripts easy access to a files meta-data without having to worry about data conversion. In addition to the basic types, arrays of all four basic types are also available.

The following three XML files give an example of the meta-data needed for processing a single image.

image XML file:

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<data>
  <Parent>../project.xml</Parent>
  <Image>
    <Resolution type=’float’>200.0,4.000</Resolution>
    <Number type=’int’>7785010</Number>
    <Name type=’str’>mAII00778501</Name>
    <NameUnmasked type=’str’>AII00778501</NameUnmasked>
    <Size type=’int’>8192</Size>
    <Magnification type=’float’>50000.0</Magnification>
    <Stepsize type=’float’>5.0</Stepsize>
    <Tilt>
      <TLTAXIS type=’float’>-49.135</TLTAXIS>
      <TLTANG type=’float’>3.360</TLTANG>
      <TLTAXA type=’float’>48.979</TLTAXA>
      <TAXA type=’float’>131.070</TAXA>
      <TANGL type=’float’>3.360</TANGL>
    </Tilt>
    <Type type=’int’>4</Type>
    <Lattice type=’float’>84.474,-0.230,0.295,84.294</Lattice>
    <Defocus type=’float’>11798.0,16126.0,-6.08</Defocus>
    <Deconvolution>
      <Origin type=’float’>0.0,0.0</Origin>
      <Intensity type=’float’>1.00000</Intensity>
    </Deconvolution>
    <Untilted type=’int’>1</Untilted>
    <SpotScan type=’float’>0</SpotScan>
    <PhaseOrigin type=’int’>1</PhaseOrigin>
  </Image>
  <Mrc>
    <TempFactor type=’float’>0.000</TempFactor>
    <Merge>
      <ZstarWindow type=’float’>0.50000</ZstarWindow>
      <PhaseResidualLimit type=’float’>50.000</PhaseResidualLimit>
      <NrOfImages type=’int’>1</NrOfImages>
      <RefineTiltGeometry type=’float’>0</RefineTiltGeometry>
    </Merge>
    <JobA>
      <Hole type=’int’>2</Hole>
      <Mask type=’int’>20</Mask>
      <Box type=’int’>400,1200</Box>
      <QuadserchRadius type=’int’>9</QuadserchRadius>
      <CcThreshold type=’float’>0.130</CcThreshold>
      <PredictRange type=’int’>7</PredictRange>
    </JobA>
    <JobB>
      <Mask type=’int’>22</Mask>
      <Box type=’int’>300,1000</Box>
      <QuadserchRadius type=’int’>8</QuadserchRadius>
      <CcThreshold type=’float’>0.170</CcThreshold>
      <PredictRange type=’int’>7</PredictRange>
    </JobB>
    <JobB2>
      <Mask type=’int’>24</Mask>
    </JobB2>
    <SynRefFlag type=’int’>3</SynRefFlag>
    <SynRefJobA>
      <TemperatureFactor type=’float’>-100.0</TemperatureFactor>
      <Hole type=’int’>20</Hole>
      <QuadserchRadius type=’int’>10</QuadserchRadius>
      <PredictRange type=’int’>7</PredictRange>
    </SynRefJobA>
    <SynRefJobB>
      <TemperatureFactor type=’float’>-150.0</TemperatureFactor>
      <Hole type=’int’>30</Hole>
      <QuadserchRadius type=’int’>5</QuadserchRadius>
      <PredictRange type=’int’>7</PredictRange>
    </SynRefJobB>
    <AQP2Flag type=’float’>0</AQP2Flag>
    <Flags>
      <KeepTemporaryFiles type=’float’>0</KeepTemporaryFiles>
    </Flags>
    <Radlim type=’float’>35.0,35.0,0.00</Radlim>
    <LastScript type=’float’>0</LastScript>
    <TTFCor1st type=’float’>0</TTFCor1st>
    <ResMaxCFTPlot type=’float’>0.33</ResMaxCFTPlot>
    <Phasecontrast type=’float’>0.93</Phasecontrast>
    <Rotate90 type=’int’>1</Rotate90>
    <RevertHK type=’float’>0</RevertHK>
    <Rotate180 type=’float’>0</Rotate180>
    <PermutateAB type=’float’>0</PermutateAB>
  </Mrc>
  <Image2>
    <Lattice type=’float’>0.0,0.0,0.0,0.0</Lattice>
  </Image2>
  <Diffraction>
    <Origin type=’float’>1.0,1.0</Origin>
  </Diffraction>
</data>
```

project XML file:

```
<?xml version=’1.0’ encoding=’UTF-8’?>
<data>
  <Parent>../microscope.xml</Parent>
  <Crystal>
    <RealUnitCell type=’int’>98.0,98.0,140,90.0</RealUnitCell>
  </Crystal>
```

```
    <Symmetry type='str'>p1</Symmetry>
  </data>
```

XML file with microscope data:

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <Microscope>
    <Cs type='float'>2.0</Cs>
    <HighVoltage type='float'>200.0</HighVoltage>
  </Microscope>
</data>
```

3.4.16 GUI

Even though XIMDISP provides a lot of functionality its use is restricted to displays using an 8 bit color depth. In addition, interactive zooming and direct manipulation of a displayed lattice or contrast transfer function is not implemented. These restrictions lead to the decision to implement a modular image viewer in IPLT for the visual representation of image and meta data.

The viewer consist of a canvas for displaying the image or diffraction pattern and several overlays for displaying different types of meta data. For example, an overlay can display a lattice, a peak list, a CTF or a high or low pass filter. For the lattice overlay, the lattice viewer originally written by J. Hebert was incorporated. From the list of overlays displayed one overlay is always defined as active and therefore gets the input focus and can accordingly act upon mouse or keyboard events. So far the overlay viewer is implemented in C++, the python wrapper, which enables the user to use the viewer from an interactive python environment, is under development.

3.4.17 Checkpointing

I incorporated a checkpointing mechanism for the python scripts used for processing. After every successful step of processing, a status file is updated, stating the current checkpoint. If a script is interrupted manually or by an error in one of the executables, the image processing can be continued at the last checkpoint after the source of interruption is eliminated.

3.5 Iplt–Image Processing Library and Toolkit for the Electron Microscopy Community

Contribution

The work presented here was published 2003 in the Journal of Structural Biology, volume 144, pages 4 to 12. It describes the IPLT image processing library and toolkit which provides the basic framework which was used to implement the algorithms used for processing of the diffraction patterns of AQP2 and SoPIP2;1. Ansgar Philippsen who is the main author of the publication is the main developer of IPLT. He implemented the core library including the plugin mechanism for input-output and the concept of the modular algorithms.

My contribution to this publication, as well as the continued development, consisted in analysis of the problems present in the current implementation of the different image processing tools, mainly focusing on MRC, the review of the IPLT architecture and interfaces in terms of feasibility, and the design of an optimized process flow by providing the experience I acquired during processing different projects using the MRC software. In addition, I implemented selected image and diffraction processing algorithms, as mentioned in the previous sections, tested and compared the novel algorithm to the established algorithms in MRC to access quality and speed improvements. I was also responsible for the IRIX port of MRC, the integration of MRC/CCP4 into IPLT, a considerable part of the newly implemented geometry library added to IPLT and the implemented edge detection and Fourier filter algorithms (see 3.5.2).

3.5.1 Summary

The IPLT image processing and library toolkit is a modular framework for the electron microscopy community. It is written in modern C++ using object oriented paradigms. The framework consists of several independent modules, communicating over defined interfaces. The core module, containing the image representation, is augmented by a series of orthogonal algorithm modules, using a powerful algorithm object concept. The IPLT library can be accessed by either C++ programs directly or by python script using the python wrapper, implemented with help of the boost-python framework. The IPLT toolkit is designed as collaborative framework focusing on the easy extendability by users at all levels. Consequently, the IPLT source code is developed under an open source license.

3.5.2 Addendum

The IPLT image processing library and toolkit is actively developed and extended to provide a steadily increasing set of algorithms and modules for the 2D electron crystallography community. For this reason part of the information contained in the publication is outdated. The FFT and CONVOLUTE methods were removed from the image handle interface and implemented as algorithms to provide a more coherent interface. The tvmet dependency was removed by adding a newly implemented geometry library, which is better cross platform compatible. The new geometry library contains, in addition to the basic vector functions (dot product, cross product, etc.) also geometric objects such as planes, points and lines with the corresponding geometric functions (distance between two objects, intersection point/line calculation, etc.).

The low level image handle implementation was also altered to give direct access to the image state implementation should the need for speed enforce it.

In addition, a whole plethora of standard image processing algorithm were implemented, for example Sobel and Canny edge detection algorithms, and Gaussian, Fermi and Butterworth high-, low- and bandpass filters.

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Structural Biology 144 (2003) 4–12

Journal of Structural Biology

www.elsevier.com/locate/yjsbi

Iplt—image processing library and toolkit for the electron microscopy community

Ansgar Philippsen,* Andreas D. Schenk, Henning Stahlberg, and Andreas Engel

Maurice Müller Institute for Structural Biology, Biozentrum, Klingelbergstr. 70, 4056 Basel, Switzerland

Received 4 June 2003, and in revised form 28 July 2003

Abstract

We present the foundation for establishing a modular, collaborative, integrated, open-source architecture for image processing of electron microscopy images, named *iplt*. It is designed around object oriented paradigms and implemented using the programming languages C++ and Python. In many aspects it deviates from classical image processing approaches. This paper intends to motivate developers within the community to participate in this on-going project. The *iplt* homepage can be found at `http://www.iplt.org`.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Electron microscopy; Image processing; Collaborative software

1. Introduction

The extent of structural information acquired by electron microscopy techniques depends on the sample quality, the instrument and the data analysis. Progress is inherently coupled to advancements in each of these areas. This contribution concentrates on the third one.

The electron microscopy (EM) community has produced a large collection of sophisticated image processing tools and pioneering mathematical approaches (for a summary of available software see http://3dem.ucsd.edu). Several powerful software packages, a number of them originating from the 1960s, are utilized today, and some continue to undergo cycles of reconstruction and extension. From a software architecture point of view, most of these packages follow what we would like to call the "classical" approach to image processing: individual executables solving specific tasks are generated, usually based on some underlying image processing library. Some sort of scripting mechanism chains the executables together to form an abstraction layer.

There is nothing inherently wrong with this approach, on the contrary, it has proven itself within this scientific community and others. Nevertheless, we would like to point out the following drawbacks:

The user is faced with several packages, each using different command syntax, formats, definitions, installation procedures, etc. As a consequence, time is spent in learning technicalities instead of producing results.

The developer that is confronted with the task of implementing a novel algorithm is limited to two choices: add code to an existing package, or write from scratch (we omit the third possibility of one person begging the other to implement a specific feature). Both possibilities have obvious disadvantages: The first one requires the availability and in-depth understanding of the source-code; the second one necessitates the implementation of many standard routines, so that only a fraction of the coding is devoted to the actual novelty. And both cases require the developer to have attained a certain programming skill level.

The aforementioned software packages are developed by a small number of specialized laboratories and extended by others to solve specific problems. While the creative potential is most obviously present, it is unfortunate that the EM community has not been able to concentrate its efforts on creating a common software platform akin to those available in X-ray crystallography (most prominently CCP4 (CCP4, 1994), CNS (Brunger et al., 1998), and lately PHENIX (Adams et al., 2002)).

---

* Corresponding author. Fax: +41-61-267-2109.
E-mail address: ansgar.philippsen@unibas.ch (A. Philippsen).

1047-8477/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jsb.2003.09.032

We therefore propose to establish an open-source software tool for the 3D EM community that is build by a collaborative effort of the community itself. To this end, we have designed a novel software architecture we deem suited for this approach. This paper presents its design and initial implementation, and it is to be understood as an invitation to community members to participate in its future development.

The recent emergence and success of open-source software (http://www.opensource.org, http://www.sf.net), such as Linux (http://www.kernel.org), Apache (http://www.apache.org), GIMP (http://www.gimp.org), or similar projects of the medical imaging community (http://www.itk.org) and X-ray crystallography community (Adams et al., 2002), serve as examples on how *iplt* is meant to develop.

We are fully aware that several open-source image processing libraries are already available; we hope to convince the critical reader that our integrated approach justifies a design and implementation "from scratch", as will be discussed at the end of this paper.

2. Design criteria

1. The system should appeal to novice and expert alike, providing several levels of user interaction and multiple possibilities for user contributions. The learning curve should be as linear as possible, requiring little investment from the novice to get started, yet enabling the expert to adapt the system to particular requirements.
2. The system should provide a core set of functionality (the *base*) that is easily extendable by *modules*. How modules interact with the base and how new modules can be implemented are most important aspects of the design.
3. A comprehensive set of documentation must be available at every level, comprising source-code documentation, usage manuals, as well as prepared examples and tutorials.
4. The system should work on all three major operating systems in use (Linux and other Unix variants, Microsoft Windows and Apple OSX).
5. To make *iplt* an open source project, an online platform (*web portal*) needs to be established and maintained. This platform contains the documentation and a repository of scripts, modules and other material; it also allows exchange of ideas and feedback on the software.

3. Roadmap

The project may be roughly divided into three phases:

(i) Evaluation of existing software tools, design, and initial implementation of the new architecture (completed).
(ii) Implementation of the most important existing algorithms and procedures to make the system usable in a working environment; involvement of community developers; setup of web-portal. This is the major focus for the coming months.
(iii) Routine utilization; implementation of novel methods to advance the field. This is the ultimate goal and vision.

This paper marks the end of the first phase (i) and the beginning of second. The challenge that the project thus faces is reminiscent of the chicken-and-egg problem: the new software will only be used if algorithms are implemented, but they will only be implemented if the system is really worth to invest time in. Our group will focus on the implementation of established and novel electron crystallography algorithms.

4. Current state of implementation

4.1. Design components

In the implementation of the aforementioned design criteria 1. and 2., we have chosen to strictly follow object oriented paradigms throughout the construction of *iplt*. Two languages were selected, namely C++ (Stroustrup, 1997) and Python (http://www.python. org). They form an ideal combination: C++ is a compiled, efficient, strongly type-checked language, Python complements with a simple yet powerful syntax, interpreter style, an easy plugin mechanism that supports either Python code or a shared library, a wide community and a plethora of components and add-ons. Both languages enjoy widespread popularity and an ever-growing user base.

The overall software architecture layout is shown in Fig. 1, comprising several layers of encapsulation and abstraction.

At the heart lies the base class library, which encapsulates the essential data representations, such as an image or a function, and helper classes, such as size, point, vector. In addition, the abstract classes that are used for algorithms and gui components are also defined here.

The interface of the base class library is available to the three components that extend it, namely the algorithms (*alg*), the graphical user interface (*gui*), and the IO plugins, each explained in its own section further below.

The next abstraction layer is delivered by a layer of wrappers, which map a specified set of C++ classes and their interface to Python, giving rise to the three main Python modules `iplt`, `iplt.alg`, and `iplt.gui`.

These modules integrate seamlessly with standard Python and thus enable further abstraction layers, written in Python themselves.

We denote by *algorithm* an independent, C++ based, exported Python module. A *procedure* is a combination of algorithms at the level of Python.

4.2. Base

4.2.1. Image encapsulation

The central class, and indeed the most complex one, is the image. It encapsulates the actual discrete values as well as the image state (using the *state* pattern (Gamma et al., 1994)). This image state is characterized by the following:

Extent: encoded by a start- and end-index, specifies the size, offset from origin, and dimensionality. An index is always an integer triple, therefore the extent defines the dimensionality of an image: 1D (width, height = depth = 1), 2D (width, height, depth = 1) or 3D (width, height, depth).

Domain: is either spatial or frequency. Several discriminations are based on this property, such as the direction of the FFT (see below) or the calculation of pixel scale.

Type: one of real, complex, or half-complex, the underlying states encode all values in double precision. (It should be pointed out that the encapsulation of the image state allows implementation of other precisions, if required).

The image class interface comprises the set of methods available to interact with an image object and includes the following functionality:

GetValue, *SetValue:* based on an index (an integer triplet), a value can be either directly set or retrieved. For half-complex data, the complex conjugate is automatically returned for values falling in the "other half."

GetIntpolValue: an interpolated value is returned based on a vector (a float triplet); the coordinate system is the same as for the indexed version above, e.g., a vector of (1.4, 3.1, −0.9) will return an interpolated value from the index block (1, 3, −1) (2, 4, 0).

FFT: a fast Fourier transform is applied to the data; the direction is given by the domain: spatial data is forward transformed, frequency data backward. It can optionally be performed using a memory efficient algorithm in-place transform.

Convolute: the image is convoluted with another image, a function or a kernel. It can be optionally performed in-place for memory efficiency reasons.

Transform: apply a linear transformation (such as rotation or scaling) to an image. The abstract class Transformer can be used to derive new transformations.

Subimage: a region indicated by a start- and end-index is returned as a new image object.

The public interface of the image class is kept minimalistic on purpose. In contrast to other image processing class libraries, it does not contain a long list of methods that implement various algorithms and procedures. Instead, it uses two specific object oriented patterns (Gamma et al., 1994) to interact with algorithms and gui elements: *visitor* and *observer*, as explained in the algorithms resp. gui section below.

Access to consecutive image values is provided by means of an iterator, which encapsulates the explicit start- and end-index.

4.2.2. Encoding meta-data

A recurring problem in any image processing application is the diversity of image formats, in particular the additional information describing an image, usually residing in the header of the file. We call this additional information meta-data, and—for the sake of discussion—have divided it into the following categories:

(i) minimal set required to read and store the image into memory: size, origin and pixel-encoding.
(ii) meta-data concerning a single, isolated image, such as acquisition date, instrument parameters or sample parameters.
(iii) meta-data concerning the processing procedure, such as batch number, processing history, assigned characteristics.

Due to the diverse and ever-changing nature of this meta-data, it is not useful to define a fixed encoding (i.e., a fixed image format). Instead, we have devised the following encapsulation scheme in our class library: The minimal information is part of the image class itself, in fact this is the defining characteristics mentioned above. The remaining meta-data is represented by a separate info object, which is organized in a tree-like hierarchy (Fig. 2): branches designate specific groups, and the leafs contain the actual data items. Each image object has a data info object associated with it, allowing the contained meta-data to be queried and modified.

The current info implementation wraps the groups and items around an XML-DOM representation (http://www.w3.org/XML), allowing direct retrieval from and storage to an XML file.

This does not entirely remove the conflict of different image formats and different conventions. It has just shifted the discussion from "which byte represents which feature" to "how should we name this feature." We propose to base this naming convention on the one introduced by the EBI for their EMDEP project (Tagari et al., 2002).

4.3. IO plugins

To actually deal with the multitude of various image formats, a plugin mechanism has been implemented, which enables the base class library to be extended with specific routines that can read and/or write a specific image format and convert it to/from the internally used representation. The read/write operation is not limited to local disks, of course, but can be performed over the network as well.

Each IO plugin must use the data-info class to convert to/from the specific image format it implements.

4.4. Algorithms

The algorithmic component is a collection of modules that extract information from an image and/or modify it. One can distinguish four different types of algorithms:

- extraction only, without input parameters (e.g., statistics calculation)
- extraction only, input parameters required (e.g., peaks search)
- modification, without input parameters (e.g., autocorrelation)
- modification, input parameters required (e.g., CTF correction)

To incorporate all of these possibilities using a single syntax, the following scheme has been devised, given an existing image object (depending on the nature of the algorithm, point 2 and/or 4 might be skipped):

1. creation of an algorithmic object (an instance of an algorithmic class)
2. setting of parameters (via algorithm object interface)
3. application to image (via image object interface)
4. extraction of results (via algorithm object interface)

A code example is given in Fig. 3 (left-hand side) to illustrate this scheme.

As mentioned above, the interaction between an algorithm and an image is implemented using the *visitor* pattern (Gamma et al., 1994): To this end, the algorithmic class—instantiated in 1.—is always derived from the visitor class (part of the base class library). This visitor class defines an interface that the algorithm must implement. The image uses this interface polymorphically to apply the algorithm to itself.

5. GUI

The graphical user interface is the third major component of *iplt*. It is implemented using wxWindows (http://www.wxwindows.org), ensuring cross-platform

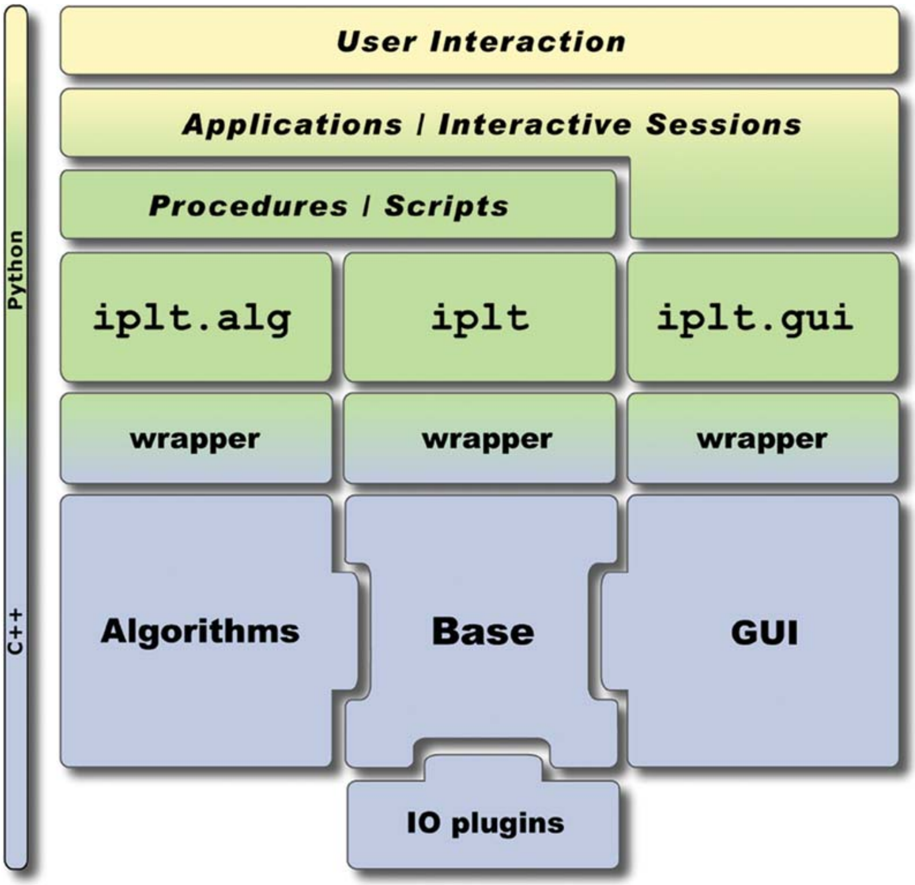


Fig. 1. Layout of *iplt*. At the heart lies the base class library, which is extended by algorithms, IO plugins and the GUI. Wrappers reflect the C++ classes into Python, in the form of three Python modules. Upon these modules, scripts, procedures, and applications written in Python are layered, with which the user interacts.

compatibility. It is not mandatory, meaning that the base and algorithmic components will function perfectly without it.

To facilitate the GUI, the *observer* pattern (Gamma et al., 1994) has been implemented in the image class. A GUI element that is designed to display image data (such as a 2D image viewer) must be derived from an observer class (part of the base class library). This observer class defines an interface that each derived class must implement. This allows a derived class to register with an image and to receive status update notifications. From the image point of view, it “knows” that one or more observers are watching it, but it has no information on the specific implementations of each observer; this is again made possible by the use of polymorphism.

It must be pointed out that the *architecture* to implement a GUI is finished, but not a fully functional GUI itself: at the moment, the 2D image viewer will simply display a given image and allow zooming, translation and individual pixel picking; it serves more as a “proof of concept” than anything else.

5.1. C++ to Python wrappers

The seamless transition between C++ and Python is ensured by the boost.python library, which is part of the boost project (http://www.boost.org). In contrast to other tools that provide similar functionality, the interaction between C++ and Python is encoded explicitly, yet straightforwardly, in C++. Subsequent compilation of such wrapper code leads to the dynamically loadable Python module. This approach has been successfully used in a novel set of tools for X-ray crystallography (Grosse-Kunstleve et al., 2002).

An example of how this applies to *iplt* is given in Fig. 3: On the one hand, boost.python maps C++ classes and their interfaces to corresponding Python classes. On the other hand, it transparently converts Python objects back to C++ pointers or references, thus providing dynamic, run-time dependent interaction between C++ based objects.

This capability is especially exploited for the interplay between image class and algorithmic modules: An image object is unaware of any specific algorithmic implementation, it does however “know” how to handle a generic visitor class. A specific algorithmic implementation is loaded as a Python module, an algorithmic object is constructed within Python, and passed to the Apply method of an image object. The underlying C++ code evaluates the algorithm object as a polymorphic pointer, thereby calling the correct code within the specific algorithmic implementation. As dictated by object-oriented concepts, the algorithms implementation contains both the methods on how to apply itself to an image, as well as the parameters and results.

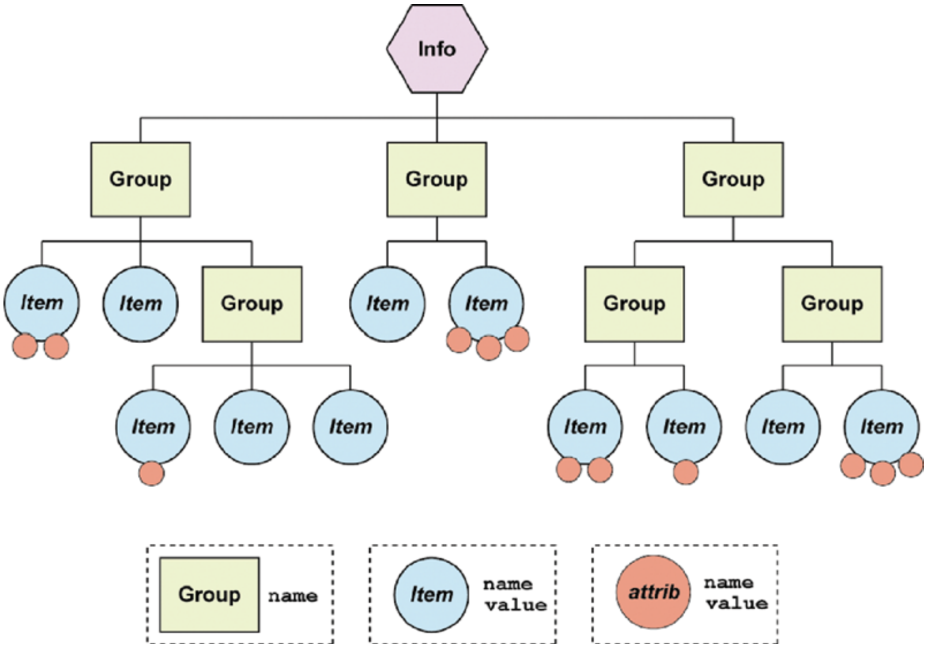


Fig. 2. The schematic layout of the meta info class. It contains a collection of groups and items, arranged in a tree-like hierarchy. A group is characterized by a name and may contain several subgroups and items. An item is characterized by a name and value, it may contain several attributes, which are also characterized by a name and value.

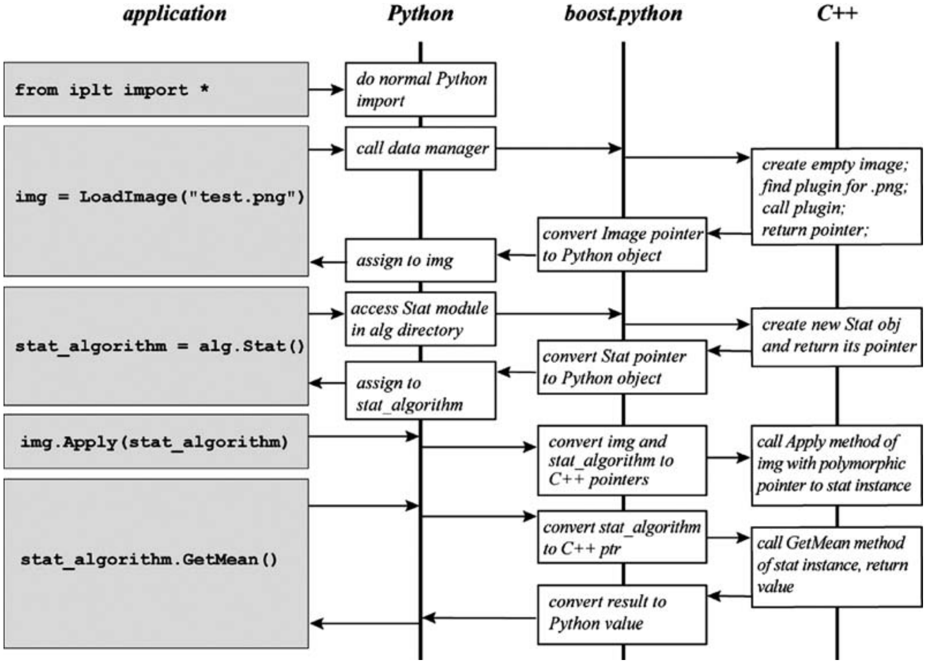

application
Python
boost.python
C++
from iplt import *
do normal Python import
img = LoadImage("test.png")
call data manager
create empty image; find plugin for .png; call plugin; return pointer;
convert Image pointer to Python object
assign to img
stat_algorithm = alg.Stat()
access Stat module in alg directory
create new Stat obj and return its pointer
convert Stat pointer to Python object
assign to stat_algorithm
img.Apply(stat_algorithm)
convert img and stat_algorithm to C++ pointers
call Apply method of img with polymorphic pointer to stat instance
stat_algorithm.GetMean()
convert stat_algorithm to C++ ptr
call GetMean method of stat instance, return value
convert result to Python value

Fig. 3. Code example and the complex interplay between Python, C++ and its wrapper, mediated by boost.python. The wrapper is responsible for converting Python objects to and from C++ pointers. From top to bottom: (i) The *iplt* package is imported (internal Python command). (ii) An image is loaded from a file and stored in a Python object called "img." (iii) A new instance of a specific algorithm is created (in this case destined to calculate some statistics), stored in "stat_algorithm." (iv) The algorithm instance is applied to the image instance "img." (v) The algorithm instance is used to retrieve results.

5.2. Building process

Two issues concerning the build process must be mentioned, both of which are addressed by the tool we have chosen to use, namely SCons (http://www.scons.org):

The aim for cross-platform compatibility is ideally extended to the building tool itself: SCons is setup in a way to automatically detect the platform and available compiler, the building rules specified within a SCons definition file are abstract.

A developer must maintain two different types of source within the *iplt* tree: the files from the distribution and his own code. The usage of SCons allows unproblematic coexistence of these two source types, integrating the custom code seamlessly within the build and install process.

6. Information exchange

6.1. Proposed documentation

There are several sets of documentation that will be offered along with the software:

The *design guide* is the fundamental document. It explains the philosophy behind the project, it defines guidelines and rules for programming style, it lists and explains the software libraries and tools used, and it reflects the vision for future development. This is work in progress and a first version should be released around the same time as the publication of this paper.

The so called *in-source documentation* is assembled automatically from specially formatted comments and the object oriented hierarchy intrinsically defined in the source code. It is aimed primarily at developers who wish to contribute algorithms or new graphical user interface elements. In essence, it reflects the information embedded in the source-code, presented in a more accessible way. It is available from the web-portal and routinely updated from the source-code.

The *standards document* will contain detailed description of the EM specific definitions and conventions used throughout the project, as well as a description of the various image formats implemented with plugins. This is work in progress and subject to change based on future discussions among the project participants; a first draft, however, is being prepared, based on conventions circulated within the community (D. Belnap, personal communication; R. Marabini, personal communication).

The *user manual* will provide a comprehensive overview of the system and serves as an entry point for new users, offering a somewhat non-technical introduction and several code examples. The writing of this manual is postponed until the project has matured.

6.2. The web portal

The project homepage can be found at `http://www.iplt.org`. At the time of writing (July 2003), it offers access to the source-code and some documentation, in particular the automatically generated in-source docs, detailed instructions for building and installation, and guides for adding new algorithms and IO-plugins. In addition, it contains information on the official mailing list, which will serve as the first medium to facilitate user and developer exchange.

7. Discussion

We have designed and implemented a novel software architecture for 3D EM image processing, with the goal to establish the first truly collaborative, open-source software project in this field. This architecture, as presented in Fig. 1, is a carefully crafted interplay of separate components and abstraction layers. It reflects our pursuit of object oriented paradigms, which are rooted in the choice of C++ and Python as the two programming languages that comprise the system.

The approach presented herein allows our main design goals to be met: (a) Provisions are made for several levels of user interaction, from the GUI to the use of ready-made Python scripts to the direct interaction with the main Python modules. (b) Community members may contribute in multiple ways and on several levels; the abstraction layers ensure a relative independence and flexibility of these contributions. (c) The incorporation of algorithm modules and I/O plugins is straightforward and allows flexible extension of the base functionality.

We chose to implement the underlying C++ image processing class library from scratch, well aware that several existing open-source libraries are available (such as http://www.itk.org). Our rational for this path was and is the optimal realization of our design criteria, in particular the interplay with Python and the implementation of algorithm objects, which necessitated this seemingly drastic measure.

The Python language is becoming more and more popular in many different fields, including the structural biology community: in X-ray crystallography, the PHENIX project (Adams et al., 2002) incorporates Python in a similar way to *iplt* (and in fact served as an inspiration); in the 3DEM community, at least one software package (EMAN, Ludtke et al., 1999) features a fully functional Python interface. The most prominent features of Python, in our opinion, are the versatility as both scripting and programming language, incorporation of object-oriented paradigms, massive amount of various modules, and easy extendibility with C/C++.

We hope to largely eliminate the problems and difficulties stemming from the variety of image file formats by the implementation of IO plugins and the encapsulation of meta-data (as shown in Fig. 2). One of the remaining issues is the comprehensive storage of the internal meta-data representation on the filesystem; with the current implementation, a separate XML file must be used to capture the full extent of the meta-data, since the commonly used file formats will only store a subset. A remedy might be the incorporation of the HDF5 file format (http://hdf.ncsa.uiuc.edu/HDF5/), which allows arbitrary data and meta-data to be stored within a single file.

The separation between the image and its algorithmic modules might seem cumbersome at first. In comparison to the more "classical" approach of implementing each possible algorithm as a method in the image interface, we would like to note the following advantages of our design: (a) Algorithms can be added to the system without recompiling the base component, they are independent entities. (b) The concept of *algorithmic objects* introduces a flexible and intuitive way to deal with the variety of potential algorithms: their individual parameters can be set or queried at any time, they can be re-used on several images, thereby even accumulating results, and they allow the various algorithm types to be incorporated using a single, comprehensive syntax. (c) The algorithm developer is forced to use the image interface, thus allowing the internals of the image class to be modified if necessary.

The graphical user interface forms an intricate, yet clearly separate component of the architecture. It is build upon wxWindows (http://www.wxwindows.org), a powerful cross-platform toolkit. Its independence from the rest of the code is established by implementing it both as a separate component as well as running it in a separate thread. As a consequence, it remains a non-mandatory component that is nevertheless tightly integrated should graphical user interaction be required.

The dependence of the *alg* and *gui* components on the base component is unidirectional: While subclasses of a visitor (algorithms) or observer (gui) exist, their specific implementation is irrelevant to the base. Therefore, changes in an algorithm or the GUI do not affect the base class library. On the other hand, both the algorithms (visitors) and gui (observer) heavily use the base class library and thus changes in the base class library potentially affect any algorithmic or gui implementation.

The collaborative aspect is manifold, following the open source philosophy "every user can contribute". The possible list of contributions includes providing constructive feedback, requesting novel features, submitting own procedures, scripts, tips, and tricks through the web portal, incorporating more algorithms, tweaking existing code, and extending the core functionality by working on the architectural foundation.

The specifications and standards adopted by *iplt* are not explicitly mentioned here, mainly because they are subject to change based upon opinions and decisions emerging from the discussion among contributors.

The current code does not include any specific parallelization features. This fact does not impede or even prohibit the later introduction of such features, because the architecture allows parallelization to be implemented at two very different levels: (a) In the context of C++, the internal functionality of classes in the base or algorithm module can be modified to utilize multiple processors (such as the FFT method of the image class). As long as the interfaces remain unchanged—and this can be expected for parallelization—these modifications do not require any adjustment outside the class. We would like to defer such parallelization to a time when the code has reached full functionality and optimization is performed. (b) In the context of Python, the iterative application of algorithmic sequences (procedures) over a potentially large number of individual images can be distributed in a multi-node or grid environment. At the moment, *iplt* has not yet reached the stage where procedures have been written and are routinely applied; as a consequence, the actual implementation of this sort of parallelization must be kept in mind until then.

The short-term goals are dictated by the roadmap, as outlined above: the next project phase marks the beginning of the actual collaboration; as a consequence, our energy is now invested in the enhancement of the web-portal, the completion of varies pieces of documentation, and of course implementation of algorithms.

To summarize, we have taken a first step in setting up a collaborative image processing tool for the 3D electron microscopy community. Its success depends on both our continuing contribution as well as the acceptance within the community. This work may raise sufficient motivation for interested laboratories to invest some initial time and energy, with the potentially rewarding outcome: a novel software tool that has emerged by the creative effort of the community itself.

8. Software tools and methods

8.1. Introduction to object oriented design

We would like to briefly mention several termini technique from that field in order to facilitate the discussion. For comprehensive treatment of this subject, the reader is referred to the wide range of excellent textbooks on this matter, such as (Eckel, 1995; Meyer, 1997; Stroustrup, 1997).

In an object oriented model, both the data itself as well as the *methods* (functions) that act on that data are combined into a *class*; this is called *encapsulation*, implicating that the data items themselves are not directly accessed, but rather through their class methods. This set of methods is called the *interface* of the class. As a consequence, the intricate detail of data organization and encoding are shielded from the "outside" (the program), and the only exposed component is the interface. An instance of a class during runtime is usually called an *object*.

The principle of code-reusage is very important in object oriented design. To this end, classes may become part of other classes data by what is called *composition*, or they may be subclassed by a mechanism called *inheritance*. The latter is additionally empowered by a feature called *polymorphism*, which enables a subclass to assume the identity of its base (parent) class, yet override some or all of the base classes methods.

8.2. Software and libraries

All software tools and libraries that are used within *iplt* are cross-platform compatible, (they are supported at least on Linux, Microsoft Windows, and Apple OSX), their source-code is available at no cost, and they allow incorporation into open-source software.

All C++ code is Standard C++ (ISO/IEC 14882) compliant. The STL (Standard Template Library (Josuttis, 1999)) and the Boost library (http://www.boost.org) are utilized as much as possible.

The Python version that is currently employed is 2.2 (http://www.python.org).

To facilitate refactoring (Fowler, 2000), unit tests are implemented using CppUnit (http://cppunit.sf.net), and are run after each build.

The GUI component is based on wxWindows (http://www.wxwindows.org).

Fast Fourier transform routines are from the fftw library (Frigo and Johnson, 1998, http://www.fftw.org).

In-source documentation is parsed and formatted with doxygen (http://www.doxygen.org).

The XML-DOM implementation is done with Xerces from the Apache project (http://www.apache.org).

Vector and Matrix classes are from tvmet (http://tvmet.sf.net).

The building process is controlled by SCons, a python-based tool (http://www.scons.org).

Source code revisions are handles by CVS, the Concurrent Version System (http://www.cvshome.org/).

Acknowledgments

This work was supported by the M. E. Müller-Foundation of Switzerland, the Swiss National Foundation

(Grant No. NF 31-59415.99) and the Swiss National Center of Competence in Research (NCCR) ‘Structural Biology’.

References

Adams, P.D., Grosse-Kunstleve, R.W., Hung, L.-W., Ioerger, T.R., McCoy, A.J., Moriarty, N.W., Read, R.J., Sacchettini, J.C., Sauter, N.K., Terwilliger, T.C., 2002. PHENIX: building new software for automated crystallographic structure determination. Acta Cryst. D 58, 1948–1954.

Brunger, A.T., Adams, P.D., Clore, G.M., DeLano, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, N., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T., Warren, G.L., 1998. Crystallography and NMR system (CNS): a new software system for macromolecular structure determination. Acta Cryst. D 54, 905–921.

Collaborative Computational Project, Number 4, 1994. The CCP4 Suite: Programs for Protein Crystallography. Acta Cryst. D 50, 760–763.

Eckel, B., 1995. Thinking in C++. Prentice Hall, Englewood Cliffs, NJ. ISBN 0-13-917709-4.

Fowler, M., 2000. Refactoring: Improving the Design of Existing Code. Addison Wesley, Reading, MA. ISBN 0-201-48567-2.

Frigo, M., Johnson, S.G., 1998. FFTW: an adaptive software architecture for the FFT. In: Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing 3, pp. 1381–1384.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1994. Design Pattern: Elements of Reusable Object-Oriented Software. Addison Wesley, Reading, MA. ISBN 0-201-63361-2.

Grosse-Kunstleve, R.W., Sauter, N.K., Moriarty, N.W., Adams, P.D., 2002. The computational crystallography toolbox: crystallographic algorithms in a reusable software framework. J. Appl. Cryst. 35, 126–136.

Josuttis, N.M., 1999. The C++ Standard Library: A Tutorial and Reference. Addison Wesley, Reading, MA. ISBN 0-201-37926-0.

Ludtke, S.J., Baldwin, P.R., Chiu, W., 1999. EMAN: semiautomated software for high-resolution single-particle reconstructions. J. Struct. Biol. 128, 82–97.

Meyer, B., 1997. Object Oriented Software Construction, second ed. Prentice Hall, Upper Saddle River, NJ. ISBN0-13-629155-4.

Stroustrup, B., 1997. The C++ Programming Language, third ed. Addison Wesley, Reading, MA. ISBN 0-201-88954-4.

Tagari, M., Newman, R., Chagoyen, M., Carazo, J.M., Henrick, K., 2002. New electron microscopy database and deposition system. Trends Biochem. Sci. 27 (11), 589.

Chapter 4

Improvements in Sample Preparation and Image Acquisition

Contribution

My contribution to the sample preparation consisted in developing an optimized protocol for trehalose embedding in collaboration with M. Chami.

The protocol is based on the protocol developed Gyobu et al.[17] using a symmetric carbon sandwich technique. Our additions mainly consist in the humidity chamber constructed in our lab and the hydraulic loop holder for applying a second carbon film, although also the incubation time and trehalose concentration were varied from the original protocol to optimize the result obtained. In addition, we developed a protocol for glucose embedding in dependence on the protocol for trehalose embedding, incorporating the idea of evaporating a second carbon layer on top of the grid, as it was proposed for samples without carbon support by Brink et al.[5].

Concerning image and diffraction data acquisition, I was responsible for the magnification calibration, the installation of the Spot Scan hardware, and the modification of the Spot Scan software, originally written by H. Stahlberg, to get equally illuminated spots. In addition, I installed the hardware responsible for controlling the drum scanner and wrote the script for the 16 bit export and numbering of diffraction patterns recorded on the CCD.

4.1 Specimen Preparation

For recording tilted images of AQP2 and SoPIP2;1 the samples were prepared using a modified version of the backinjection technique[21] applying a second carbon layer as described in Gyobu et al.[17]. In addition a protocol for embedding samples in glucose was developed inspired by the work done in 1998 by Brink et al.[5] for samples without carbon film support.

4.1.1 Trehalose Embedding

A little petri dish is filled to the top with water. Carbon film is floated off the mica onto the water. The carbon film is carefully fished out with a 400[1] mesh molybdenum grid[2] which is beforehand plunged several times into water, in order to make it more hydrophilic. The grid with the carbon film is subsequently lowered onto three drops of 8% trehalose[3], deposited in the cleft of a M-shaped parafilm to exchange the water droplet on the film with trehalose solution. The trehalose droplet on the grid is removed with a pipette and only a small amount of about ~1 μl is left on the grid. The grid is put upside down into a humid chamber–constructed out of two petri dishes (see Fig. 4.1) to prevent the evaporation of the droplet. 3 μl[4] of sample is back injected and pipetted several times up and down to ensure proper absorption of the sample onto the carbon film. Typical adsorption times are 2 min for AQP2 and 10 min for SoPIP2;1.

A second carbon film is floated on water. It is fished out with a platinum loop which is mounted on a hydraulic holder (Fig. 4.2), by slowly lowering the loop onto the carbon film until it reaches the water surface. When the loop is raised the carbon film will swim on a thin water film contained in the loop, due to the water tension. The loop is then slowly lowered onto the prepared grid to depose the second carbon film. The grid is blotted from two sides with Watmann filter paper until the grid is translucent. Afterwards, the grid is quickfrozen in liquid nitrogen (Fig. 4.3). During freezing, the trehalose prevents formation of crystalline ice, which would destroy the protein crystal.

For recording untitled images, tilted images in Spot Scan mode, or diffraction patterns

---

[1] 200-300 mesh grids can also be taken to increase recordable area at the cost of reduced carbon film flatness.

[2] Pacific Grid Tech, 2115 Greencove Lane, Sugar land, TX 77479, USA,`http://www.grid-tech.com`

[3] The trehalose concentration should be adjusted to according to the air humidity of the preparation room. A dry surrounding favors evaporation of water and therefore leads to a concentration of the used trehalose solution. Due to the higher evaporation rate given when the second layer is omitted 2% trehalose should be taken for single layered preparations.

[4] depending on the concentration of the sample the volume can be reduced up to 1 μl

at low tilt angle, the placement of the second carbon film can also be omitted.

4.1.2 Glucose Embedding

Samples can also be embedded in glucose instead of trehalose. The glucose protocol follows the protocol for trehalose embedding up to the blotting step. Due to the higher water retaining capability of glucose, the sample does not have to be quickfrozen prior to transferring it into the microscope. After fixing the sample to a cryo holder and mounting the holder into the microscope, the sample can be frozen inside the microscope by cooling the cryo holder with liquid nitrogen, avoiding ice contamination. Even though a sample can be more easily embedded in glucose than in trehalose, with the additional advantage of less contamination, in some cases trehalose embedding should be favored because freezing under vacuum conditions can lead to dehydration of the sample[5].

If placement of the second carbon layer is omitted, it is even possible to evaporate a carbon layer on top of the grid. The second carbon layer having a tight contact to the ice helps to divert the charge built up during illumination.

4.2 Image and Electron Diffraction Acquisition

4.2.1 Calibration

It is known that there is some difference between the nominal magnification set during image acquisition at the microscope and the real magnification achieved on the negative. Therefore, calibration of the magnification is crucial, because the size of the crystal lattice and ultimately the size of the 3D mass density is dependent on the magnification. The calibration of the CM200FEG was done by recording images of a gold cross grating. Negatives were recorded at several magnifications, and the diameter of the grid squares was subsequently measured and compared to the expected square diameter, yielding the actual magnification. As seen in Fig. 4.4, the deviation between nominal magnification and measured magnification is around 1% – 2% for the magnification range used for imaging. The deviation is higher at very high magnification. The calibration data was confirmed by measuring the length of TMV viruses (data not shown).

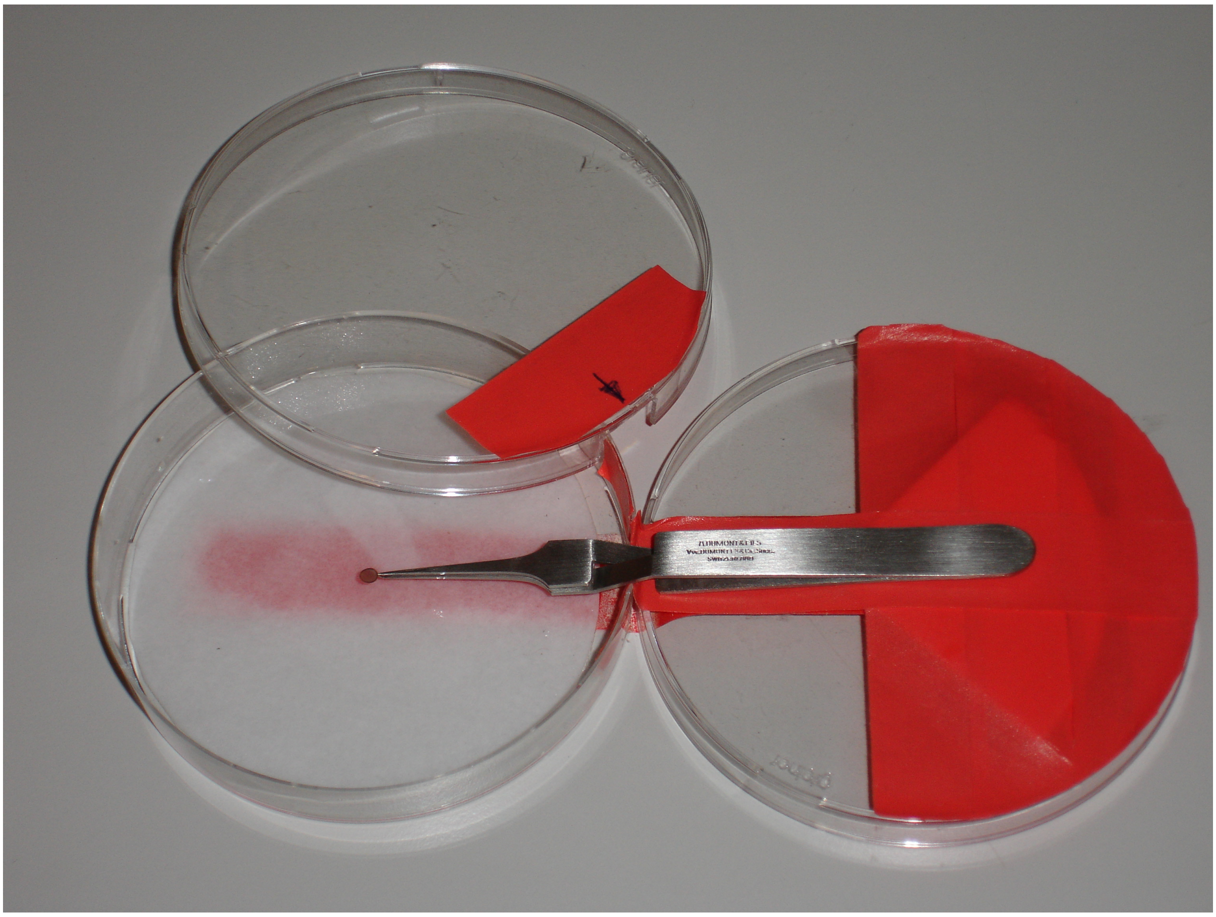


Figure 4.1: Humidity chamber constructed out of two petri dishes. One petri dish is used as a chamber with a wet filter paper at the bottom to provide a humid environment. The dish and its lid have a notch at one side leaving space for the tweezers. The lid of the second dish and a folded filter paper taped on top of the lid provide support for the tweezers. The original design was done by T. Braun. The support platform for the tweezers attached to the chamber was enlarged in my modified version to improve stabilization of the tweezers.

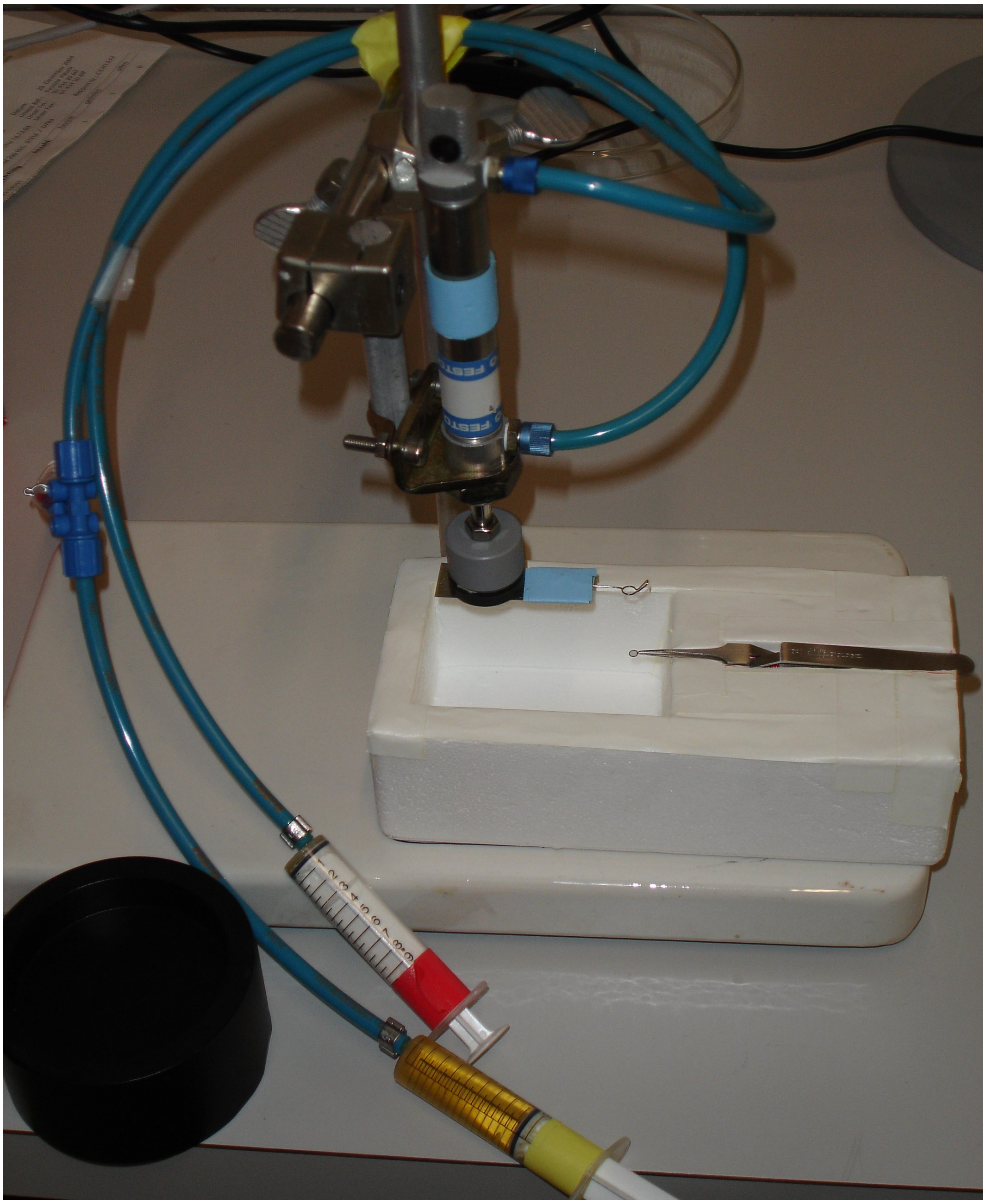


Figure 4.2: Hydraulic loop holder. The platinum loop attached by a magnet can slowly be moved up and down by the two syringes to place a floated carbon film on top of the microscopy grid with minimal force. The design of the hydraulic loop holder was realized together with M. Chami.

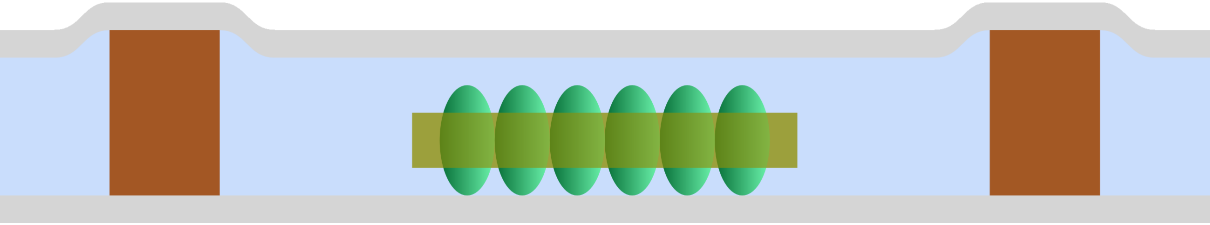


Figure 4.3: Scheme of a trehalose embedded crystal.The crystal is shown in green and the amorphous ice in blue. A second carbon layer to reduce charging is present.

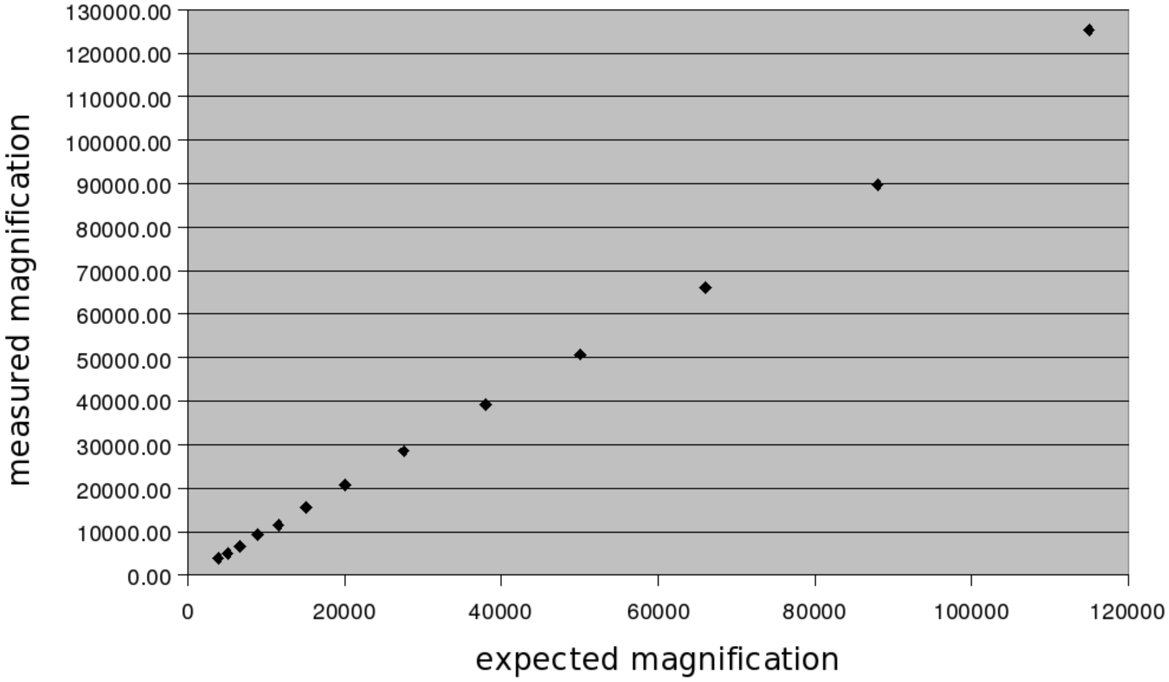
4.2.2 Spot Scan Imaging

To record a 3 dimensional dataset of a 2d crystal, the sample has to be tilted to acquire projection images of the unit cell at different tilt angles. In comparison to the recording of untilted images, there are two additional difficulties occurring for tilted image acquisition. First, the CTF of a microscope is dependant on the defocus and therefore the CTF function varies within a tilted image due to the changing distance of the sample to the focal plane. Second, the sample is charged by inelastically scattered electrons that ionize atoms in the sample, leading to a deflection of the electron beam, which results in an effect similar to thermal drift.

To reduce charging of the sample, highly tilted images can be recorded in Spot Scan mode[9]. To implement this technique an additional PC equipped with a CIO-DA02/16[5] DA-converter card was attached to the microscope. Two channels, providing an analog voltage between -5V and 5V, are connected directly to the beam deflection coils. During imaging the beam is focused to a small spot (5 mm - 10 mm on the negative) with low intensity and is moved across the sample at high interval by changing the voltage applied to the beam deflection coils. A spot on the sample is typically illuminated for 50 ms - 100 ms. The program controlling the DA-converter is implemented as a C-program within the DOS operating system. The first generation of Spot Scan images showed several darker spots after development, caused by slightly longer illumination at given positions. Interrupts raised during the illumination time proved to be the source of the unequal illumination time, and therefore deactivation of maskable interrupts during image acquisition leaded to equally illuminated spots.

---

[5] Specifications: channels:2, Output Resolution: 16 bit, Output Range ±5V, Output settling: 12μs typ; 19μs max



Expected magnification	Calculated magnification
3810	3811.76
5000	5076.00
6600	6646
8800	9204
11500	11435
15000	15480
20000	20736
27500	28485
38000	39096
50000	50760
66000	66240
88000	89640
115000	125280

Figure 4.4: Magnification calibration for the CM200FEG

4.2.3 Image Digitalization

Negatives are digitized using a Heidelberg Primescan 7100 drum scanner. Cryo micrographs of protein crystals are scanned at a resolution of 2000 lines per cm using the transmissive mode of the scanner, with the detector aperture at position -4. The NewColor scan program is configured to refocus the detector system on every scanned negative. The images are saved in TIF format using a 16 bit color depth. The scanner, which internally records greyscale values corresponding to the optical density, unfortunately converts these values to negative transmission. As a consequence, the images have to be transformed back to optical density before image processing.

4.3 Electron Diffraction

4.3.1 Acquisition

Electron diffraction patterns were recorded on a Gatan UltraScan^{TM} 2k×2k CCD mounted before a GIF-energy filter. The beam stop originally provided with the microscope was used, blocking the central beam to avoid damaging the scintillator of the CCD. The only modification done to the beam stop was the squeezing of the tip perpendicular to the beam to reduce it's cross section on the recorded CCD image (Fig. 4.5).

The CCD images are recorded on a PC equipped with the Digital Micrograph imaging software. Digital micrograph was set up in a way that TIF images can be exported with 16bit color depth. Two additional scripts were written to facilitate automatic numbering and saving of recorded electron diffraction patterns.

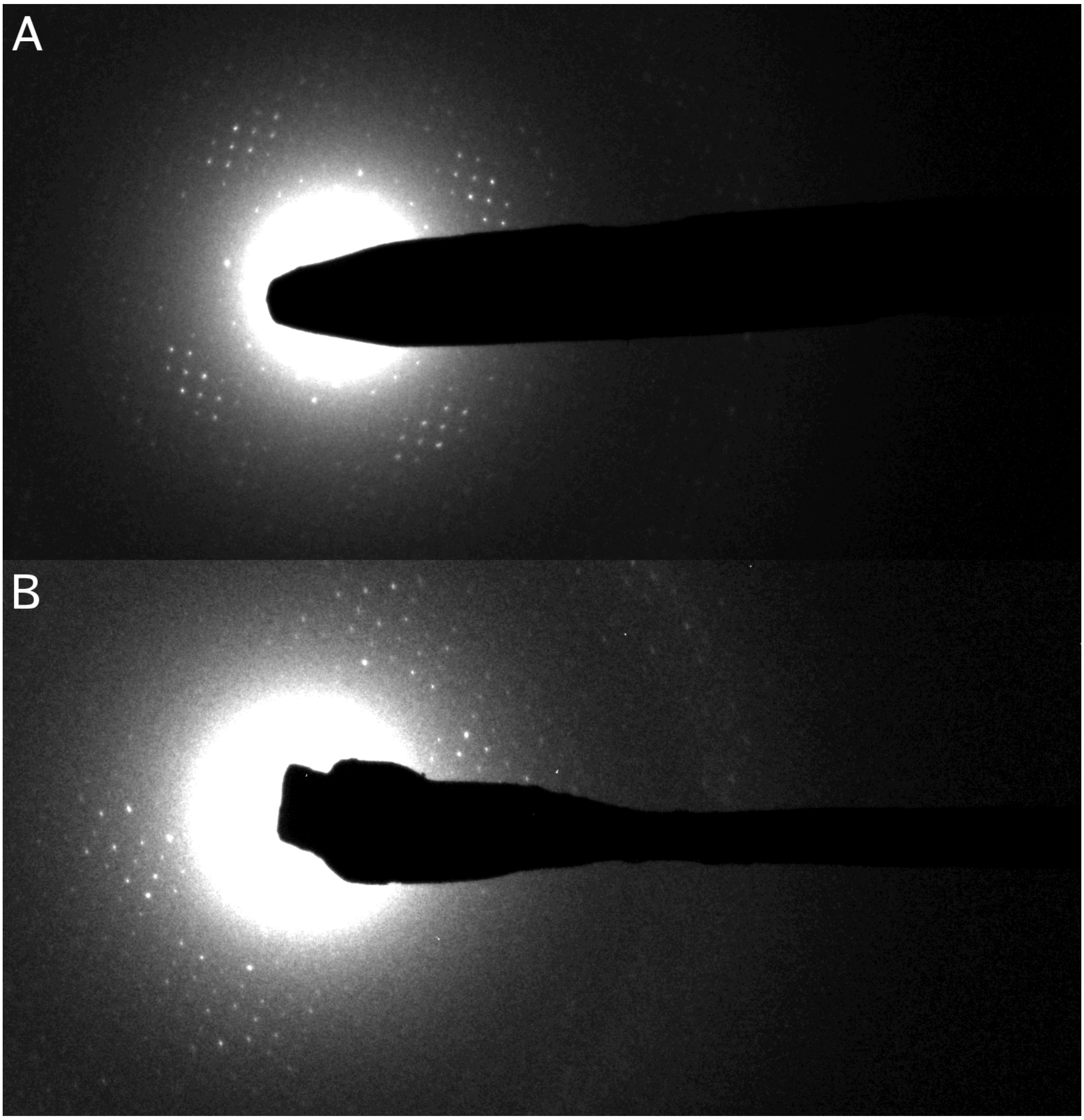


Figure 4.5: Beam stop modification. (A) unmodified beam stop (B) squeezed beam stop

Chapter 5

Conclusion

The structural analysis done on the presented aquaporin projects clearly shows that the determination of an atomic structure lies within reach, even for challenging samples as the double layered crystals, even though sufficient resolution has not yet been achieved. The development of a novel image processing concept is clearly needed to reach this goal, harvesting the strength of the different packages already available, either by integrating the existing algorithms into a new software project or by enabling the interaction between different software packages by defining and implementing common interfaces. The open source character of IPLT and its modularity hopefully attract more people contributing to the project at different levels in the future .

Integration of novel algorithms for processing electron diffraction patterns in IPLT showed promising results for AQP2, hopefully giving the improvement in resolution needed for structure determination at an atomic level.

Appendix

Abbreviations

AFM: atomic force microscope / atomic force microscopy

AMP: adenosine 5' monophosphate

AQP: Aquaporin

BCA: bicinchoninic acid

BSA: bovine serum albumin

cAMP: cyclic adenosine 3',5' monophosphate

$\mathbf{C_{12}E_8}$**:** n-dodecyl-octaethyleneglycol ether

CCD: charge coupled device

CHAPS: 3-[(3-cholamidopropyl) dimethyl-ammonio] propanesulfonic acid

CMC: critical micellar concentration

CPU: central processing unit

CTF: contrast transfer function

DA: digital-analog

DDM: n-Dodecyl-β-D-maltopyranoside

DM: n-Decyl-β-D-maltopyranoside

DMPC: Dimyristol phophatidylcholine

DOPE: dioleoyl-glycerophosphatidylethanolamine

DOPG: dioleoyl-glycerophosphoglycerol

dpi: dots per inch

EDTA: ethyl-n-diamin-tetraacetat

EM: Electron Microscope / Electron Microscopy

FFT: fast Fourier transform

Glp: Glycerol facilitator-like protein

LDAO: Lauryldimethylamine-oxide

lpc: lines per centimeter

LPR: lipid-to- protein ratio

MOI: multiplicity of infection

MPA: mass-per-area

MRC: Medical Research Council / also used as abbreviation for the image processing programs suite originating from there

MS: mass spectrometry

Ni-NTA: Nickel-nitriloacetic acid

octyl-POE: octyl-polyoxyethylene

OD: Optical density

OG: n-Octyl-β-D-glucopyranoside

OTG: n-octyl-β-D-thioglucopyranoside;

POPC: 1-palmitoyl-2-oleoyl-sn-glycero-3-phosphocholine

ppm: pixel per millimeter

RAID: redundant array of inexpensive disks

SAED: Selected area electron diffraction

SGI: Silicon Graphics

STEM: Scanning transmission electron microscope

TCIF: Tilted contrast imaging function

TED: Transmission Electron Diffraction

TEM: Transmission electron microscope / Transmission electron microscopy

TTF: tilted transfer function

TX100 Triton X-100

Bibliography

[1] HSL: A collection of Fortran codes for large scale scientific computation. http://www.numerical.rl.ac.uk/hsl, 2004.

[2] D.A. Agard. A least-squares method for determining structure factors in three-dimensional tilted-view reconstructions. *J Mol Biol*, 167(4):849–52, 1983.

[3] N. Blot, C. Berrier, N. Hugouvieux-Cotte-Pattat, A. Ghazi, and G. Condemine. The oligogalacturonate-specific porin KdgM of Erwinia chrysanthemi belongs to a new porin family. *J Biol Chem*, 277(10):7936–44, 2002.

[4] W. Boos and H. Shuman. Maltose/maltodextrin system of Escherichia coli: transport, metabolism, and regulation. *Microbiol Mol Biol Rev*, 62(1):204–29, 1998.

[5] J. Brink, H. Gross, P. Tittmann, M.B. Sherman, and W. Chiu. Reduction of charging in protein electron cryomicroscopy. *J Microsc*, 191 (Pt 1):67–73, 1998.

[6] G. Chandy, G.A. Zampighi, M. Kreman, and J.E. Hall. Comparison of the water transporting properties of MIP and AQP1. *J Membr Biol*, 159(1):29–39, 1997.

[7] F. Chaumont, F. Barrieu, E. Wojcik, M.J. Chrispeels, and R. Jung. Aquaporins constitute a large and highly divergent protein family in maize. *Plant Physiol*, 125(3):1206–15, 2001.

[8] P.M. Deen and N.V. Knoers. Physiology and pathophysiology of the aquaporin-2 water channel. *Curr Opin Nephrol Hypertens*, 7(1):37–42, 1998.

[9] K.H. Downing. Spot-scan imaging in transmission electron microscopy. *Science*, 251(4989):53–9, 1991.

[10] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".

[11] J. Garnier, J.F. Gibrat, and B. Robson. GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol*, 266:540–53, 1996.

[12] C. Geourjon and G. Deleage. SOPM: a self-optimized method for protein secondary structure prediction. *Protein Eng*, 7(2):157–64, 1994.

[13] T. Gonen, Y. Cheng, P. Sliz, Y. Hiroaki, Y. Fujiyoshi, S.C. Harrison, and T. Walz. Lipid-protein interactions in double-layered two-dimensional AQP0 crystals. *Nature*, 438(7068):633–8, 2005.

[14] T. Gonen, P. Sliz, J. Kistler, Y. Cheng, and T. Walz. Aquaporin-0 membrane junctions reveal the structure of a closed water pore. *Nature*, 429(6988):193–7, 2004.

[15] N. Grigorieff. Three-dimensional structure of bovine NADH:ubiquinone oxidoreductase (complex I) at 22 A in ice. *J Mol Biol*, 277(5):1033–46, 1998.

[16] Y. Guermeur, C. Geourjon, P. Gallinari, and G. Deleage. Improved performance in protein secondary structure prediction by inhomogeneous score combination. *Bioinformatics*, 15(5):413–21, 1999.

[17] N. Gyobu, K. Tani, Y. Hiroaki, A. Kamegawa, K. Mitsuoka, and Y. Fujiyoshi. Improved specimen preparation for cryo-electron microscopy using a symmetric carbon sandwich technique. *J Struct Biol*, 146(3):325–33, 2004.

[18] R. Henderson, J. Baldwin, K. Downing, J. Lepault, and F. Zemlin. Structure of purple membrane from halobacterium halobium: recording, measurement and evaluation of electron micrographs at 3.5 A resolution. *Ultramicroscopy*, 19:147–178, 1986.

[19] R. Henderson, J.M. Baldwin, T.A. Ceska, F. Zemlin, E. Beckmann, and K.H. Downing. Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *J Mol Biol*, 213(4):899–929, 1990.

[20] R. Henderson and P.N. Unwin. Three-dimensional model of purple membrane obtained by electron microscopy. *Nature*, 257(5521):28–32, 1975.

[21] T. Hirai, K. Murata, K. Mitsuoka, Y. Kimura, and Y. Fujiyoshi. Trehalose embedding technique for high-resolution electron crystallography: application to structural study on bacteriorhodopsin. *J Electron Microsc (Tokyo)*, 48(5):653–8, 1999.

[22] Y. Hiroaki, K. Tani, A. Kamegawa, N. Gyobu, K. Nishikawa, H. Suzuki, T. Walz, S. Sasaki, K. Mitsuoka, K. Kimura, A. Mizoguchi, and Y. Fujiyoshi. Implications of the Aquaporin-4 Structure on Array Formation and Cell Adhesion. *J Mol Biol*, 355(4):628–639, 2006.

[23] R.D. King and M.J. Sternberg. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Sci*, 5(11):2298–310, 1996.

[24] M.A. Knepper, J.G. Verbalis, and S. Nielsen. Role of aquaporins in water balance disorders. *Curr Opin Nephrol Hypertens*, 6(4):367–71, 1997.

[25] R. Koebnik, K.P. Locher, and P. Van Gelder. Structure and function of bacterial outer membrane proteins: barrels in a nutshell. *Mol Microbiol*, 37(2):239–53, 2000.

[26] W. Kuhlbrandt, D.N. Wang, and Y. Fujiyoshi. Atomic model of plant light-harvesting complex by electron crystallography. *Nature*, 367(6464):614–21, 1994.

[27] E.R. Kunji, S. von Gronau, D. Oesterhelt, and R. Henderson. The three-dimensional structure of halorhodopsin to 5 A by electron crystallography: A new unbending procedure for two-dimensional crystals by using a global reference structure. *Proc Natl Acad Sci U S A*, 97(9):4637–42, 2000.

[28] S.P. Mallick, B. Carragher, C.S. Potter, and D.J. Kriegman. ACE: automated CTF estimation. *Ultramicroscopy*, 104(1):8–29, 2005.

[29] K. Murata, K. Mitsuoka, T. Hirai, T. Walz, P. Agre, J.B. Heymann, A. Engel, and Y. Fujiyoshi. Structural determinants of water permeation through aquaporin-1. *Nature*, 407(6804):599–605, 2000.

[30] S. Nielsen, J. Frokiaer, and M.A. Knepper. Renal aquaporins: key roles in water balance and water balance disorders. *Curr Opin Nephrol Hypertens*, 7(5):509–16, 1998.

[31] H. Nikaido. Porins and specific diffusion channels in bacterial outer membranes. *J Biol Chem*, 269(6):3905–8, 1994.

[32] Y. Noda and S. Sasaki. Trafficking mechanism of water channel aquaporin-2. *Biol Cell*, 97(12):885–92, 2005.

[33] T. Pellinen, H. Ahlfors, N. Blot, and G. Condemine. Topology of the Erwinia chrysanthemi oligogalacturonate porin KdgM. *Biochem J*, 372(Pt 2):329–34, 2003.

[34] A. Philippsen, A.D. Schenk, H. Stahlberg, and A. Engel. Iplt–image processing library and toolkit for the electron microscopy community. *J Struct Biol*, 144(1-2):4–12, 2003.

[35] Philippsen A., Engel H.A., and Engel A. The Contrast Imaging Function for Tilted Specimens. *Ultramicroscopy*.

[36] Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. *Numerical recipes in Fortran 77*. Cambridge University Press, 2003.

[37] G.M. Preston, T.P. Carroll, W.B. Guggino, and P. Agre. Appearance of water channels in Xenopus oocytes expressing red cell CHIP28 protein. *Science*, 256(5055):385–7, 1992.

[38] Reimer L. *Transmission Electron Microscopy*. Springer, 1997.

[39] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol*, 232(2):584–99, 1993.

[40] P. Shaw and G.J. Hills. Tilted Specimen in the Electron Microscope: A simple Specimen Holder and the Calculation of Tilt Angles for Crystalline Specimens. *Micron.*, 12:279–282, 1981.

[41] V.W. SIDEL and A.K. SOLOMON. Entrance of water into human red cells under an osmotic pressure gradient. *J Gen Physiol*, 41(2):243–57, 1957.

[42] S. Tornroth-Horsefield, Y. Wang, K. Hedfalk, U. Johanson, M. Karlsson, E. Tajkhorshid, R. Neutze, and P. Kjellbom. Structural mechanism of plant aquaporin gating. *Nature*, 2005.

[43] P.N. Unwin and R. Henderson. Molecular structure determination by electron microscopy of unstained crystalline specimens. *J Mol Biol*, 94(3):425–40, 1975.

[44] G. Valenti, G. Procino, G. Tamma, M. Carmosino, and M. Svelto. Minireview: aquaporin 2 trafficking. *Endocrinology*, 146(12):5063–70, 2005.

[45] T. Walz, T. Hirai, K. Murata, J.B. Heymann, K. Mitsuoka, Y. Fujiyoshi, B.L. Smith, P. Agre, and A. Engel. The three-dimensional structure of aquaporin-1. *Nature*, 387(6633):624–7, 1997.

Curriculum Vitae

General information

Last Name: Schenk
First Name: Andreas Daniel

Personal details

Date of Birth: 05.01.1977
Nationality: Swiss
Sex: male

Education

2002-2006: PhD. studies in the group of Prof. Andreas Engel, M.E. Müller Institute for Microscopy, University of Basel
2001: Diploma degree Biologie II, University of Basel
1997-2001: Undergraduate studies Biologie II, University of Basel
1993-1996: Matura Mathematics/Physics (Typus C), Gymnasium Muttenz
1989-1993: Progymnasium Muttenz
1984-1989: Primary school Muttenz

Practical experience

2002-2006: PhD. studies in the group of Prof. Andreas Engel, M.E. Müller Institute for Microscopy, University of Basel
Title: **Structure Determination of Membrane Proteins by Electron Crystallography**
2001: One week workshop Electron Crystallography school Barcelona
2000-2001: Nine month Diploma thesis in the group of Prof. Andreas Engel, M.E. Müller institute for Microscopy, University of Basel
Title: **Aquaporin 2: purification crystallization and high resolution electron microscopy**.

Languages

German: mother tongue
English: Good knowledge
French: Good knowledge

Publications

The 5Å structure of heterologously expressed plant aquaporin SoPIP2;1
Kukulski W, Schenk AD, Johanson U, Braun T, de Groot BL, Fotiadis D, Kjellbom P, Engel A.
J Mol Biol. 2005 Jul 22;350(4):611-6.

The 4.5 Å structure of human AQP2
Schenk AD, Werten PJ, Scheuring S, de Groot BL, Müller SA, Stahlberg H, Philippsen A, Engel A.
J Mol Biol. 2005 Jul 8;350(2):278-89.

Iplt--image processing library and toolkit for the electron microscopy community
Philippsen A, Schenk AD, Stahlberg H, Engel A.
J Struct Biol. 2003 Oct-Nov;144(1-2):4-12.

Membrane protein reconstitution and crystallization by controlled dilution
Remigy HW, Caujolle-Bert D, Suda K, Schenk A, Chami M, Engel A.
FEBS Lett. 2003 Nov 27;555(1):160-9.

Presentations

2004:	Action meeting Kopenhagen: Structure determination of AQP2
2003:	Aquaplugs meeting Arhuus: Structure determination of AQP2 2nd Meeting of the NCCR membrane protein groups Villingen: Progress in the structure of AQP2
2002:	Aquaplugs meeting Hamburg:Progress report on the structure determination of Aquaporin 2 Aquaplugs/Action meeting Nendaz: Structure determination of Aquaporin 2
2001	Aquaplugs meeting Edinburgh:, Aquaporin 2

Selected Posters

Method Development in 2D Electron Crystallography
Schenk AD, Philippsen A,Signorell GA,Remigy HW,Chami M,Werten PJL,Kukulski W, Engel A
Biozentrum Annual Symposium, October 2005; Basel, Switzerland.

The waterchannel Aquaporin2 (AQP2)
Schenk AD, Philippsen A,Signorell GA,Engel A, Werten PJL
FEBS-IUBMB Conference, July 2005, Budapest, Hungary

The 5 Å Structure of Heterologously Expressed Plant Aquaporin SoPIP2;1
Kukulski W, Schenk AD,Johanson U,Braun T, de Groot BL, Fotiadis D, Kjellbom P, Engel A
FEBS-IUBMB Conference, July 2005, Budapest, Hungary

Structure determination of two aquaporins by electron crystallography
Schenk AD, Kukulski W, Braun T, Karlsson M, Kjellbom P, Fotiadis D, Werten PJL, Stahlberg H, Scheuring S, Engel A
Biozentrum Annual Symposium, October 2004; Basel, Switzerland.

The 3D Structure of Aquaporin-2 elucidated by Cryo EM
Schenk AD, Werten PJL, Philippsen P, Stahlberg H, Engel A
2nd international conference on Structure, Dynamics and Function of Proteins in Biological Membranes , Oktober 2003, Monte Verita, Switzerland

2D-Crystallization and Electron Crystallography of the Plant Aquaporin PM28A
Kukulski W, Schenk AD, Braun T, Karlsson M, Kjellbom P, Fotiadis D, Engel A
2nd international conference on Structure, Dynamics and Function of Proteins in Biological Membranes, Oktober 2003, Monte Verita, Switzerland

The 3D structure of aquaporin-2 elucidated by cryo EM.
Werten PJL, Schenk AD, Stahlberg H, Engel A
Gordon Research conference on 3D electron microscopy. Colby-Sawyer College, June 2003, New London, NH, USA

HTEX: high throughput electron crystallography (an automated approach).
Schenk AD, Philippsen A, Werten PJL, Braun T, Engel A, Stahlberg H
Gordon Research Conference on 3D Electron Microscopy. Colby-Sawyer College, June 2003, New London, NH, USA

The waterchannel AQP2; A structural Study Combining AFM and Cryo-EM
Schenk A, Frederix P, Scheuring S, Philippsen A, Engel A, Stahlberg H
Biozentrum Annual Symposium, 2002; Basel, Switzerland.

Academic and community service

2000-present:	Member and Webmaster of the "Junges Forum Gentechnologie" (www.jufogen.ch)

Computer skills

Skills

Programming skills:	C++, C, Fortran, Basic, Pascal, Python, PHP, ColdFusion, Actionscript, html
Operating Systems	
-Administration:	Linux, IRIX, Windows (3.1,95,98,NT4,ME,2000,XP), Mac OS (9,X), DOS, OS2,
-User experience:	VMS, Solaris,OS/9, QNX, BeOS, Netware
Database skills:	SQL, Access
Office skills:	Word, Excel, Power Point ,Corel Draw, Corel Photopoint, Adobe Photoshop, Open Office, Latex, Lyx, Macomedia Flash
Server skills:	Apache, MySQL, SAMBA, FTP, NFS, Conference Room, BSCW

Courses

2004:	Flash MX 2004 extension course: Universitätsrechenzentrum University of Basel
2002:	Flash MX course: Universitätsrechenzentrum University of Basel ColdFusion course: Universitätsrechenzentrum University of Basel
1996:	C-course: GIB Muttenz

Miscellaneous

2000-present:	Employee Universitätsrechenzentrum University of Basel
2003:	reassignment to Swiss Army Upper Cadre Training facility
1998:	employee Zürcher Kantonalbank (stock exchange, risk controlling) (one month)
1997:	employee Zürcher Kantonalbank (stock exchange, risk controlling) (three month) military service: Swiss Airforce Intelligence (4 month)